

# WXP Program Reference

Version 5.0

31 July, 1998

|                |     |
|----------------|-----|
| CLIMAT.....    | 1   |
| CONTOUR.....   | 3   |
| DOMTRAN.....   | 7   |
| FO_PARSE.....  | 9   |
| FOCALC.....    | 14  |
| FORECAST.....  | 23  |
| FOUSWX.....    | 26  |
| GIFCAT.....    | 35  |
| GRBCALC.....   | 36  |
| GRBSND.....    | 42  |
| GRDMATH.....   | 51  |
| GRIBLOOK.....  | 59  |
| GRID.....      | 63  |
| HDRPARSE.....  | 68  |
| HEAT.....      | 70  |
| HPGLCAT.....   | 72  |
| HURRICANE..... | 73  |
| INGEST.....    | 77  |
| LIGHT.....     | 87  |
| LOOPSET.....   | 91  |
| MAPPLT.....    | 92  |
| MC2AREA.....   | 97  |
| MOIST.....     | 98  |
| PARSE.....     | 99  |
| PRFPLT.....    | 103 |
| PSCAT.....     | 107 |
| RAD.....       | 109 |
| RADCVT.....    | 118 |
| RAWMATH.....   | 121 |
| SA_PARSE.....  | 127 |
| SACVT.....     | 133 |
| SCOUR.....     | 137 |
| SFCCALC.....   | 138 |
| SFCWX.....     | 145 |
| SHEFCVT.....   | 153 |
| SMCVT.....     | 157 |
| STATLOG.....   | 162 |
| STDATMS.....   | 169 |
| SUNCALC.....   | 171 |
| TAFCVT.....    | 174 |
| UA_PARSE.....  | 177 |
| UACALPLT.....  | 182 |
| UACVT.....     | 190 |
| UNIT.....      | 193 |
| UPAIRWX.....   | 194 |
| UPCALC.....    | 202 |
| VECTOR.....    | 209 |
| WCHILL.....    | 213 |
| WXP.....       | 215 |
| WXPCITY.....   | 219 |

|              |     |
|--------------|-----|
| WXPDEF.....  | 222 |
| WXPFILE..... | 224 |
| WXPLOOP..... | 227 |
| XSAT.....    | 234 |

# CLIMAT

## NAME

**climat** - CLIMAT data conversion program

## SYNOPSIS

**climat** [*parameters...*] *filename*

## PARAMETERS

| Command Line                             | Resource                  | Default             | Description   |
|--|---------------------------|---------------------|---|
| <b>-h</b>                                | <a href="#">help</a>      | No                  | Lists basic help information.   |
| <b>-df=filename</b>                      | <a href="#">default</a>   | .wxpdef             | Sets the name of the resource file.   |
| <b>-na=name</b>                          | <a href="#">name</a>      | climat              | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                               | <a href="#">batch</a>     | No                  | Run program in batch mode   |
| <b>-me=level</b>                         | <a href="#">message</a>   | out2                | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>decoded output -<b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>                      | <a href="#">file_path</a> | current directory   | Specifies location of database files.   |
| <b>-dp=datapath</b>                      | <a href="#">data_path</a> | current directory   | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.  |
| <b>-nc=name_conv</b>                     | <a href="#">name_conv</a> | name_conv           | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>                       | <a href="#">in_file</a>   | cli_dat             | Specifies the input file name tag. The default is <b>cli_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-of=out_file</b>                      | <a href="#">out_file</a>  | None                | Specifies the output file name tag. If this is not specified, the output is to standard output. The tag is used to create the output file name.   |
| <b>-ou=output</b>                        | <a href="#">output</a>    | text,g111           | The <b>output</b> resource specifies the output of the program. <ul style="list-style-type: none"> <li><b>text</b> - text output</li> <li><b>raw</b> - output ready for use in raw file (use <b>out_file</b> to save to file)</li> <li><b>data</b> - raw data</li> <li><b>gall</b> - decode all groups</li> <li><b>g111</b> - decode only group 1</li> <li><b>g222</b> - decode only group 2</li> <li><b>g333</b> - decode only group 3</li> <li><b>g444</b> - decode only group 4</li> </ul> |
| <b>-cu=[hour la]</b>                     | <a href="#">current</a>   | None                | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=hour</b>                          | <a href="#">hour</a>      | None                | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-pa=param</b><br>[ <i>,param...</i> ] | <a href="#">parameter</a> | None                | Extra parameters  |
| <i>filename</i> [#seq]                   | <a href="#">filename</a>  | None<br>User Prompt | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-Batch: current= <b>la</b> WXP files.  |

## DESCRIPTION

The CLIMAT decoding program will search for and decode CLIMAT reports.

```
CLIMAT
05998 48455
  111 10069 20076 30314013 403590273 5334 60231416
      7253117 8000000 9000000
  222 06190 10062 20069 30292012 403000256 5316
      6022013 7216 8000000 9000000
  333 03131 12400 31612 40401 8020000
  444 0033116 1024031 2038301 3023806 4067030
      5380006 61500 711000
```

The output is either textual or raw. The textual format lists a parameter description along with its value. The raw format readies the data to be put into a raw file:

```
WXPRAW
OZ  4 JAN 98
!Climate Data
ID   StPr  SLPr   T   Tdev  TMx   TMn   VPr  Prec Q Dy Sun Pct $
60101 1016.9 1019.4 12.2  5.7  15.4   9.0 10.8  145 4  8 105 63
60115  962.6 1018.2   9.8  2.4  15.1   5.1  8.7   43 4  8 164 52
60120 1018.0 1019.7 13.1  0.3  17.0   9.1 12.2   93 4 12 191 111
60135 1011.0 1021.0 12.0  1.8  16.7   8.0 12.6  109 4 12 188 20
60150  954.9 1019.2   9.4  2.4  21.2  -1.7  9.7   90 3 13 134 73
60155 1012.7 1019.7 13.1  0.4  16.9   9.0 11.5   61 3  9 195 62
60195  849.6 1151.1   6.4  8.6  11.2   1.7  6.5   20 4  2 298 75
60220 1019.1 1019.9 15.4  0.3  19.0  11.8 14.4   34 2  5 221 93
60230  964.4 1019.5 11.8  1.0  16.7   7.3 10.4   46 4 -9 199 95
60250 1015.2 1018.0 15.4  2.2  20.3   9.8 -99   33 4  4 239 77
60265  889.1  512.1 11.2  2.7  17.9   5.1  6.2   34 5  3 268 86
60318 1018.3 1019.5 12.9  2.0  16.1   9.7 11.5  129 4 12 184 114
```

## EXAMPLES

```
climat -cu=la -of=cli_raw
```

will convert the latest CLIMAT file and create raw output file.

## FILES

## SEE ALSO

- [mapplt](#) - plots raw files
- [rawmath](#) - the raw file math program

# CONTOUR

## NAME

**contour** - Contours existing grids

## SYNOPSIS

**contour** [*parameters...*] *filename*

## PARAMETERS

| Command Line             | Resource                    | Default  | Description  |
|--------------------------|-----------------------------|--|--|
| <b>-h</b>                | <a href="#">help</a>        | No   | Lists basic help information.  |
| <b>-df=filename</b>      | <a href="#">default</a>     | .wxpdef  | Sets the name of the resource file.  |
| <b>-na=name</b>          | <a href="#">name</a>        | <b>contour</b>   | Specifies the name used in resource file parsing.  |
| <b>-ba</b>               | <a href="#">batch</a>       | No   | Run program in batch mode  |
| <b>-me=level</b>         | <a href="#">message</a>     | <b>out2</b>  | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>   |
| <b>-fp=filepath</b>      | <a href="#">file_path</a>   | current directory  | Specifies location of database files.  |
| <b>-dp=datapath</b>      | <a href="#">data_path</a>   | current directory  | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>       | <a href="#">con_path</a>    | current directory  | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>       | <a href="#">raw_path</a>    | current directory  | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>      | <a href="#">grid_path</a>   | current directory  | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>     | <a href="#">image_path</a>  | current directory  | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>     | <a href="#">name_conv</a>   | <b>name_conv</b>   | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>       | <a href="#">in_file</a>     | <b>grd</b>   | Specifies the input file name tag. The default is <b>grd</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-mf=mapfile[,...]</b> | <a href="#">map_file</a>    | <b>wxp.map</b> or<br><b>cont.dat</b> ,<br><b>country.dat</b> for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu=[hour]la</b>      | <a href="#">current</a>     | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=hour</b>          | <a href="#">hour</a>        | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=num_hour</b>      | <a href="#">num_hour</a>    | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.  |
| <b>-pd=domain</b>        | <a href="#">plot_domain</a> | User prompt<br>Batch: first  | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |

|  |                              |  |  |
|--|------------------------------|--|--|
|  |                              | domain in <b>wxp.reg</b> file.                                 |  |
| <b>-re=region</b>                      | <a href="#">region</a>       | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-pl=plot_type</b>                   | <a href="#">plot_type</a>    | User prompt<br>Batch: fill contours or vectors                 | Specifies the type of output plot. By default, the user is prompted for the plot type. To produce a contour or vector plot, use the following plot types:<br><b>Contours:</b> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>                | <a href="#">plot_scale</a>   | <b>1.0</b>   | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm=format</b>                      | <a href="#">plot_format</a>  | None   | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot_param</a>   | None   | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.  |
| <b>-dr=drawable</b><br>[,...]          | <a href="#">draw</a>         | <b>all</b>   | Specifies which part of the plot is to be drawn.   |
| <b>-la=label</b> [,...]                | <a href="#">label</a>        | Program defaults   | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-in=interval</b>                    | <a href="#">con_interval</a> | None for plots,<br>User prompt for grids. Batch: <b>0</b>      | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.   |
| <b>-cb=base</b>                        | <a href="#">con_base</a>     | Lowest contour value, maximum wind speed                       | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <b>con_base</b> resource.  |
| <b>-fl=font_list</b>                   | <a href="#">font_list</a>    | <b>modern.fnt</b>  | Specifies a list of fonts to use in the plots.   |
| <b>-ct=color_table</b>                 | <a href="#">color_table</a>  | <b>wxp.clr</b>   | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |
| <b>-cot=color[:attr]</b>               | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the text color for labels above and below the plot.  |
| <b>-cod=color[:attr]</b>               | <a href="#">color_data</a>   | <b>green</b>   | Specifies the color for plotted data.  |
| <b>-com=color[:attr]</b>               | <a href="#">color_map</a>    | <b>red</b>   | Specifies the color for background maps.   |
| <b>-coln=color[:attr]</b>              | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>  | Specifies the color for lat/lon lines.   |
| <b>-cocm=color[:attr]</b>              | <a href="#">color_cmap</a>   | <b>black</b>   | Specifies the color for color fill contour maps.   |
| <b>-coco=color[:attr]</b>              | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the color for line contours.   |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color_fill</a>   | A predefined color set based on the <b>wxp.clr</b> color table | Specifies the color fill contour color set. See the <b>color_fill</b> resource for more details.   |
| <b>-cola=color[:attr]</b>              | <a href="#">color_label</a>  | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the line contour label color.  |
| <b>-cocl=color[:attr]</b>              | <a href="#">color_clabel</a> | <b>black</b>   | Specifies the color fill contour label color.  |

|  |                            |   |  |
|--|----------------------------|---|--|
| <b>-de=device</b> [,...]   | <a href="#">device</a>     | User prompt<br>Batch: <b>display</b>            | Specifies the output device.   |
| <b>-ti=title</b>   | <a href="#">title</a>      | Program name                                    | Specifies the window title.  |
| <b>-ic</b>   | <a href="#">icon</a>       | <b>Off</b>                                      | Specifies whether to open window in iconified mode.  |
| <b>-bg=color</b>   | <a href="#">background</a> | <b>black</b><br><b>white</b> for inverse plots  | Specifies the background color of the window.  |
| <b>-ge=geometry</b>  | <a href="#">geometry</a>   | <b>640x480</b>                                  | Specifies the window and paper geometry.   |
| <i>filename</i> [+ <i>time</i> : <i>lev</i><br><i>el</i> : <i>var</i> [: <i>units</i> ]] | <a href="#">filename</a>   | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the surface converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. |

## DESCRIPTION

This program contours existing grid files either created by WXP or preformatted GRIB files. WXP grid files can be created with the **out\_file** resource being set on the contouring programs such as **sfccalc** and **upcalc**. For multigrid files, additional information can be tacked onto the filename to denote which grid to use. If not specified on the command line, the filename will be prompted for. The filename syntax is:

```
filename[+time:level:var[:units]]  
or  
filename[#num[:units]]
```

where *filename* is the name of the grid file to use. If a relative path is used, the value of the **grid\_path** is prepended to the filename. The filename can be "**std**" where standard input is used.

The extra information following the "+" or "#" defines which grid to use within a multigrid file. The value of *num* represents the sequence number of the grid in the file. Grids are numbered starting at 0 (zero). This number can be determined through a **griblook** listing.

The values of *time*, *level*, and *var* make it possible to specify the grid based on typical grid information. Possible values for each are:

- **time:** init,06,12,18,24,30,36,42,48,60,72,d3,d4,d5,d6,d7,d8,d9,d10
- **level:** sfc, sl (sea level), trop, mwnd, 1000, 850, 700, 500, 400, 300, 200, 100
- **var:** temp, rhum, pres, vvel, uwnd, vwnd, prec, lift

The *unit* specification allows the user to change the units of the grid before performing the math.

The user may also specify the input file using the [in\\_file](#) resource to specify a name convention tag and the [current](#) resource. To generate a loop, the [num\\_hour](#) resource can be specified. This will tell the program to loop through this number of hours.

The domain defaults to the domain of the first grid unless changed through the [plot\\_domain](#) resource. The domain can be interactively changed using the mouse once the plot has been generated.

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the grid file is opened and data read in. The background map will plot first unless the output is a color fill contour in which the map plots last. Next, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.

- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## EXAMPLES

```
contour -pl=cf -in=2 -de=d temp.grd
```

This produces a color fill contour plot of temperature at an interval of 2.

## FILES

## SEE ALSO

- [sfccalc](#) - the surface contouring program
- [upccalc](#) - the upper air contouring program

# DOMTRAN

## NAME

**domtran** - Domain/Projection transformation program

## SYNOPSIS

**domtran** [*parameters*] [*transform location*]

## PARAMETERS

| Command Line              | Resource                    | Default   | Description  |
|---------------------------|-----------------------------|---|--|
| <b>-h</b>                 | <a href="#">help</a>        | No  | Lists basic help information.  |
| <b>-df=filename</b>       | <a href="#">default</a>     | .wxddef   | Sets the name of the resource file.  |
| <b>-na=name</b>           | <a href="#">name</a>        | heat  | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                | <a href="#">batch</a>       | No  | Run program in batch mode  |
| <b>-me=level</b>          | <a href="#">message</a>     | out2  | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>output values - <b>print</b></li> </ul>   |
| <b>-pd=domain</b>         | <a href="#">plot_domain</a> | None<br>Batch: first domain in <b>wxp.reg</b> file. | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-re=region</b>         | <a href="#">region</a>      | Same as plot domain                                 | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-pa=parameter</b>      | <a href="#">parameter</a>   | none  | Specifies additional parameters  |
| <i>transform location</i> | <a href="#">value</a>       | none<br>User prompt                                 | <i>transform</i> - Specifies the transformation type: <ul style="list-style-type: none"> <li><b>earth2proj</b> -- transforms lat,lon to projection X,Y</li> <li><b>earth2grid</b> -- transforms lat,lon to grid X,Y</li> <li><b>proj2earth</b> -- transforms projection X,Y to lat,lon</li> <li><b>proj2grid</b> -- transforms projection X,Y to grid X,Y</li> <li><b>grid2earth</b> -- transforms grid X,Y to lat,lon</li> <li><b>grid2proj</b> -- transforms grid X,Y to projection X,Y</li> </ul> <i>location</i> - Specifies the input location either as <i>lat,lon</i> or <i>X,Y</i> . |

## DESCRIPTION

The **domtran** program is an interface into the WXP domain/projection library. The user specifies a transformation and a location and the program transforms the data to the requested coordinates. The transformation is based on the **plot\_domain**. Regions can also be specified.

```
domtran -pd=us earth2proj 40,-90
6.755947,-55.022774
```

The input is the lat,lon coordinate 40,-90. South latitude and west longitude are negative. The output is the X,Y coordinates on the projection plane. In this case the projection plane is a polar stereographic projection centered on 97W longitude.

```
domtran -pd=us earth2grid 40,-90
14.937368,7.268939
```

The output here is the location of the points relative to the grid specified by the plot domain. In this case, the grid is a 25x17 grid centered on 39,-97 with a grid spacing of 2.3 (230 km). The upper left hand gridpoint is 0,0 and the lower right gridpoint is 24,16.

## **FILES**

- **wxp.reg** - The default region file

## **SEE ALSO**

# FO\_PARSE

## NAME

**fo\_parse** - MOS data display program

## SYNOPSIS

**fo\_parse** [*parameters...*] *filename*

## PARAMETERS

| Command Line                             | Resource                   | Default   | Description  |
|--|----------------------------|---|--|
| <b>-h</b>                                | <a href="#">help</a>       | No  | Lists basic help information.  |
| <b>-df=filename</b>                      | <a href="#">default</a>    | .wxpdef   | Sets the name of the resource file.  |
| <b>-na=name</b>                          | <a href="#">name</a>       | <b>fo_parse</b>                                 | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                               | <a href="#">batch</a>      | No  | Run program in batch mode  |
| <b>-me=level</b>                         | <a href="#">message</a>    | <b>out2</b>                                     | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>   |
| <b>-fp=filepath</b>                      | <a href="#">file_path</a>  | current directory                               | Specifies location of database files.  |
| <b>-dp=datapath</b>                      | <a href="#">data_path</a>  | current directory                               | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-nc=name_conv</b>                     | <a href="#">name_conv</a>  | <b>name_conv</b>                                | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>                       | <a href="#">in_file</a>    | <b>mos_dat</b>                                  | Specifies the input file name tag. The default is <b>mos_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-cu=[hour la]</b>                     | <a href="#">current</a>    | None  | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=hour</b>                          | <a href="#">hour</a>       | None  | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-mo=model</b>                         | <a href="#">model</a>      | <b>ngm</b>                                      | Specifies the model to use as input data. <ul style="list-style-type: none"> <li><b>ngm</b> -- Use Nested Grid Model (NGM) MOS data (FOUS14) (DEFAULT),</li> <li><b>mrfx</b> -- Use Medium Range Forecast (MRF) MOS data (FOX),</li> <li><b>ngm_ua</b> -- Use upper air NGM MOS (FOUE,M,W60-78)</li> <li><b>eta_ua</b> -- Use upper air ETA MOS (FOUS60-78)</li> </ul> |
| <b>-id=identifier</b>                    | <a href="#">identifier</a> | User prompt                                     | Used to plot a specific observation.   |
| <b>-pa=param</b><br>[ <i>,param...</i> ] | <a href="#">parameter</a>  | None  | Specifies additional parameters. See the <a href="#">parameter</a> resource for more details.  |
| <i>filename</i> [ <i>#seq</i> ]          | <a href="#">filename</a>   | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the MOS data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files.   |

## DESCRIPTION

This program displays MOS (Model Output Statistics) data for a specified station. The input to the program is a raw ingested MOS data file. The program will display data from 4 different model sources:

- **NGM** - MOS statistical forecast data derived from the NGM model. This includes many parameters such as max/min temperature, 3 hourly estimates of temperature, dewpoint, precipitation, winds, etc.
- **NGM\_UA** - This is derived directly from the NGM model grids and represent data such as boundary layer temperature, relative humidity, sea level pressure, lifted index, 1000-500 mb thickness and vertical velocity.
- **ETA\_UA** - This is very similar to the NGM\_UA except the data are from the ETA model. This includes lowest level temperatures, relative humidity, sea level pressure, lifted index, 1000-500 mb thickness and vertical velocity.
- **MRFX** - This is derived extended range forecast data taken from the MRF model run once a day. The data include max/min temperature, probability of precipitation, winds, and cloud coverage reported once every 12 hours.

The model type can be changed with the [model](#) resource.

The programs starts off by prompting the user for input MOS file name. For some ingestors, such as the LDM, the list may contain many files that don't contain plottable MOS data. Only the 0 and 12Z data files are usable. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **mos\_dat** tag but this can be changed with the [in\\_file](#) resource.

**NOTE:** The program assumes that the model data for all model types are contained in a single file type represented by the **mos\_dat** tag. If the model types are in different files, the tag must be specified in the [in\\_file](#) resource. Both model and tag can be specified simultaneously if the model type is contained in the tag. For example, using **mos\_ngm\_ua** for the tag for NGM\_UA data. The program will see the "ngm\_ua" in the tag and use that model type.

Next, the program prompts the user for the station identifier. This will generally be a 3 letter identifier (US stations without the leading 'K').

Next the program will search for the particular station in the specified model data. The output of the program will vary on the model specified.

## Model NGM

NGM MOS data for IND  
Initial time: 00Z 29 SEP 97

| HOUR | TIME   | TEMP | DEW | DIR | SPD | CIL | COV | VIS | WX | MX/MN | POP12 | SNOW |
|------|--------|------|-----|-----|-----|-----|-----|-----|----|-------|-------|------|
| 6    | 29/ 6Z | 59   | 53  | 260 | 11  | 250 | BKN | 10  |    |       |       |      |
| 12   | 29/12Z | 54   | 49  | 240 | 11  | 250 | SCT | 10  |    |       |       |      |
| 18   | 29/18Z | 70   | 48  | 250 | 22  | 250 | SCT | 10  |    |       |       |      |
| 24   | 30/ 0Z | 66   | 50  | 290 | 18  | 250 | SCT | 10  |    | 73    | 21    |      |
| 30   | 30/ 6Z | 57   | 50  | 280 | 12  | 250 | BKN | 10  |    |       |       |      |
| 36   | 30/12Z | 54   | 49  | 280 | 12  | 250 | BKN | 10  |    | 52    | 30    |      |
| 42   | 30/18Z | 65   | 51  | 300 | 15  | 20  | BKN | 10  |    |       |       |      |
| 48   | 1/ 0Z  | 60   | 50  | 320 | 11  | 250 | SCT | 10  |    | 68    | 8     |      |
| 54   | 1/ 6Z  | 54   | 50  | 310 | 5   |     | CLR |     |    |       |       |      |
| 60   | 1/12Z  | 51   | 48  | 300 | 5   |     | CLR |     |    | 49    | 0     |      |

The columns are:

- HOUR - forecast hour
- TIME - valid time
- TEMP - forecasted temperature (F) at valid time
- DEW - forecasted dewpoint (F) at valid time
- DIR - wind direction (deg) at valid time
- SPD - wind speed (knt) at valid time

- CIL - cloud ceiling (100s ft) at valid time
- COV - cloud cover at valid time
- VIS - visibility (mi) at valid time
- WX - estimated weather at valid time
- MX/MN - extreme temperature (either maximum or minimum) over 12 hour period prior to valid time
- POP12 - probability of precipitation over 12 hour period prior to valid time
- SNOW - estimated snowfall during 12 hour period prior to valid time

### Model NGM\_UA, ETA\_UA

NGM MOS data for IND  
Initial time: 00Z 29 SEP 97

| HOUR   | TIME   | PSL  | PREC | DIR | SPD | THK  | LI | VERT |
|--------|--------|------|------|-----|-----|------|----|------|
| 0      | 29/ 0Z | 1004 |      | 280 | 18  | 5630 | 1  | 3.9  |
| 6      | 29/ 6Z | 1005 | 0.00 | 300 | 21  | 5550 | 3  | 1.8  |
| 12     | 29/12Z | 1005 | 0.00 | 260 | 21  | 5560 | 5  | -3.8 |
| 18     | 29/18Z | 1001 | 0.00 | 240 | 19  | 5600 | 5  | -1.2 |
| 24     | 30/ 0Z | 999  | 0.00 | 270 | 28  | 5600 | 2  | 0.6  |
| 30     | 30/ 6Z | 1002 | 0.00 | 280 | 23  | 5580 | 5  | -0.3 |
| 36     | 30/12Z | 1004 | 0.00 | 290 | 24  | 5580 | 7  | -0.9 |
| 42     | 30/18Z | 1009 | 0.00 | 310 | 12  | 5600 | 8  | -3.0 |
| 48     | 1/ 0Z  | 1013 | 0.00 | 330 | 15  | 5610 | 8  | -3.5 |
| Total: |        |      | 0.00 |     |     |      |    |      |

The columns are:

- HOUR - forecast hour
- TIME - valid time
- PSL - sea level pressure (mb)
- PREC - estimated precipitation (in) over 6 hour period prior to valid time
- DIR - boundary layer wind direction (deg) at valid time
- SPD - boundary layer wind speed (knt) at valid time
- THK - 1000-500 mb thickness (m) at valid time
- LI - 4 layer lifted index
- VERT - 700 mb vertical velocity (mb/sec)

### Model MRF

MRF MOS data for IND  
Initial time: 00Z 28 SEP 97

| DAY | DATE   | MAX | NRM      | MIN | NRM      | CLOUDS | WINDS | POP12 |
|-----|--------|-----|----------|-----|----------|--------|-------|-------|
| 0   | 28 SEP | 72  | 71 ( +1) |     | 49       | 61     | 9     | 31    |
| 1   | 29 SEP | 68  | 71 ( -3) | 51  | 49 ( +2) | 42 39  | 8 14  | 16 24 |
| 2   | 30 SEP | 67  | 71 ( -4) | 49  | 49 ( +0) | 55 68  | 9 11  | 31 25 |
| 3   | 1 OCT  | 67  | 71 ( -4) | 45  | 49 ( -4) | 44 39  | 5 7   | 17 8  |
| 4   | 2 OCT  | 73  | 71 ( +2) | 47  | 49 ( -2) | 31 41  | 4 7   | 9 12  |
| 5   | 3 OCT  | 72  | 71 ( +1) | 52  | 49 ( +3) | 37 45  | 6 8   | 14 21 |
| 6   | 4 OCT  | 73  | 71 ( +2) | 50  | 49 ( +1) | 42 52  | 7 9   | 19 22 |
| 7   | 5 OCT  | 68  | 71 ( -3) | 49  | 49 ( +0) | 50 50  | 6 10  | 25 22 |

The columns are:

- DAY - forecast day
- DATE - valid date

- MAX - maximum temperature (F) for that day
- NRM - normal maximum temperature (F) followed by departure from normal
- MIN - minimum temperature (F) for that day
- NRM - normal minimum temperature (F) followed by the departure from normal
- CLOUDS - cloud coverage (%). The first number is at 12Z and the second number is at 00Z the next day
- WINDS - wind speed (knt). The first number is at 12Z and the second number is at 00Z the next day
- POP12 - 12 hour probability of precipitation (%). The first number is at 12Z and the second number is at 00Z the next day

### Present Weather Determination

The estimated type of weather on NGM plots is based on the following information:

- Precipitation based on probability of precipitation > 30%,
- Precipitation type based on PTYPE field (rain, snow, freezing rain),
- Precipitation intensity based on quantitative precipitation QPF field (see below),
- Existence of thunderstorms based on probability of thunderstorm > 20%,
- Existence of severe thunderstorms based on severe thunderstorm probability > 30%,
- Existence of obscuration (fog/haze) based on VIS field < 1 mile where obscuration type determined by OBVIS field.

#### Precipitation type and intensity based on 6 hour information

| Type     | 1   | 2   | 3  | 4   | 5   |
|----------|-----|-----|----|-----|-----|
| Rain     | R-  | R-  | R- | R   | R+  |
| Freezing | ZR- | ZR- | ZR | ZR+ | ZR+ |
| Snow     | S-  | S-  | S  | S+  | S+  |

#### Precipitation type and intensity based on 12 hour information

| Type     | 1   | 2   | 3   | 4  | 5   | 6   |
|----------|-----|-----|-----|----|-----|-----|
| Rain     | R-  | R-  | R-  | R- | R   | R+  |
| Freezing | ZR- | ZR- | ZR- | ZR | ZR+ | ZR+ |
| Snow     | S-  | S-  | S-  | S  | S+  | S+  |

### EXAMPLES

```
fo_parse -cu=la -mo=ngm -id=IND
```

This displays NGM data for IND:

```
NGM MOS data for IND
```

```
Initial time: 00Z 29 SEP 97
```

| HOUR | TIME   | TEMP | DEW | DIR | SPD | CIL | COV | VIS | WX | MX/MN | POP12 | SNOW |
|------|--------|------|-----|-----|-----|-----|-----|-----|----|-------|-------|------|
| 6    | 29/ 6Z | 59   | 53  | 260 | 11  | 250 | BKN | 10  |    |       |       |      |
| 12   | 29/12Z | 54   | 49  | 240 | 11  | 250 | SCT | 10  |    |       |       |      |
| 18   | 29/18Z | 70   | 48  | 250 | 22  | 250 | SCT | 10  |    |       |       |      |
| 24   | 30/ 0Z | 66   | 50  | 290 | 18  | 250 | SCT | 10  |    | 73    | 21    |      |
| 30   | 30/ 6Z | 57   | 50  | 280 | 12  | 250 | BKN | 10  |    |       |       |      |
| 36   | 30/12Z | 54   | 49  | 280 | 12  | 250 | BKN | 10  |    | 52    | 30    |      |
| 42   | 30/18Z | 65   | 51  | 300 | 15  | 20  | BKN | 10  |    |       |       |      |
| 48   | 1/ 0Z  | 60   | 50  | 320 | 11  | 250 | SCT | 10  |    | 68    | 8     |      |
| 54   | 1/ 6Z  | 54   | 50  | 310 | 5   |     | CLR |     |    |       |       |      |
| 60   | 1/12Z  | 51   | 48  | 300 | 5   |     | CLR |     |    | 49    | 0     |      |

## FILES

## SEE ALSO

- [fouswx](#) - the MOS data plotting program

# FOCALC

## NAME

**focalc** - MOS data contouring program

## SYNOPSIS

**focalc** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description  |
|------------------------------|----------------------------|-------------------|--|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.  |
| <b>-df=filename</b>          | <a href="#">default</a>    | <b>.wxddef</b>    | Sets the name of the resource file.  |
| <b>-na=name</b>              | <a href="#">name</a>       | <b>focalc</b>     | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode  |
| <b>-me=level</b>             | <a href="#">message</a>    | <b>out2</b>       | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• gridding information - <b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | <b>name_conv</b>  | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | <b>mos_dat</b>    | Specifies the input file name tag. The default is <b>mos_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of raw or grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII grid or raw file</li> <li>• <b>wxpb</b> - WXP binary grid file</li> <li>• <b>raw</b> - WXP ASCII raw file</li> <li>• <b>cdf</b> - netCDF binary grid file</li> </ul> Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>mos_raw</b> or <b>mos_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type: <code>wxp:mos_grd</code><br>A full filename can be specified which can contain name convention tags. |

|   |                              |  |   |
|---|------------------------------|--|---|
| <b>-pf=</b> <i>file_param</i>   | <a href="#">file_param</a>   | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul> |
| <b>-cf=</b> <i>cityfile</i>   | <a href="#">city_file</a>    | <b>mos.cty</b>   | Specifies the name of the city database file used in finding station locations.   |
| <b>-mf=</b> <i>mapfile</i> [,...]   | <a href="#">map_file</a>     | <b>wxp.map</b> or<br><b>cont.dat</b> ,<br><b>country.dat</b> for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]   | <a href="#">current</a>      | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=</b> <i>hour</i>   | <a href="#">hour</a>         | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-mo=</b> <i>model</i>  | <a href="#">model</a>        | <b>ngm</b>   | Specifies the model to use as input data. <ul style="list-style-type: none"> <li>• <b>ngm</b> -- Use Nested Grid Model (NGM) MOS data (FOUS14) (DEFAULT),</li> <li>• <b>mrfx</b> -- Use Medium Range Forecast (MRF) MOS data (FOX),</li> <li>• <b>ngm_ua</b> -- Use upper air NGM MOS (FOUE,M,W60-78)</li> <li>• <b>eta_ua</b> -- Use upper air ETA MOS (FOUS60-78)</li> </ul>  |
| <b>-id=</b> <i>identifier</i>   | <a href="#">identifier</a>   | None   | Used to plot a specific observation. If not specified, all stations will be plotted.  |
| <b>-pd=</b> <i>domain</i>   | <a href="#">plot_domain</a>  | User prompt<br>Batch: first<br>domain in<br><b>wxp.reg</b> file.                                     | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-gd=</b> <i>grid_domain</i>  | <a href="#">grid_domain</a>  | Same as plot<br>domain   | This specifies the domain to use when fitting data to a grid. If this is not specified, it is the <b>plot_domain</b> .  |
| <b>-re=</b> <i>region</i>   | <a href="#">region</a>       | Same as plot<br>domain   | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-pr=</b> <i>prior</i>  | <a href="#">stat_prior</a>   | Depends on<br>domain size  | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.  |
| <b>-ft=</b> <i>time</i>   | <a href="#">time</a>         | User prompt<br>Batch: initial<br>time  | This specifies the forecast time to be used in the plots. To create loops, a list of forecast time are specified separated by commas.   |
| <b>-va=</b> <i>variable</i>   | <a href="#">variable</a>     | User prompt<br>Batch: first<br>variable listed in<br>the <b>sfcwx.var</b><br>file                    | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.  |
| <b>-oa=</b> <i>smooth</i><br>[, <i>radinf</i> ][, <i>passes</i> ]<br>[, <i>converg</i> ]<br>[, <i>min_sta</i> ] | <a href="#">object_param</a> | <b>1,3,1,,2,1</b>  | Specifies the objective analysis parameters. There are 5 parameters including smoothing, radius of influence, number of passes, convergence parameter and minimum number of stations.   |

|                                   |                              |  |   |
|-----------------------------------|------------------------------|--|---|
| <b>-pl=plot_type</b>              | <a href="#">plot_type</a>    | <b>Data Plot</b><br>User prompt if plotting grids<br>Batch: fill contours or vectors | Specifies the type of output plot. By default, this is a simple data plot. To produce a contour or vector plot, use the following plot types:<br><b>Contours:</b> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <b>Vectors:</b> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>           | <a href="#">plot_scale</a>   | <b>1.0</b>   | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>                 | <a href="#">plot_format</a>  | None   | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.  |
| <b>-pp=param</b><br>[,param...]   | <a href="#">plot_param</a>   | None   | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>         | <a href="#">draw</a>         | <b>all</b>   | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>            | <a href="#">label</a>        | Program defaults   | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.   |
| <b>-in=interval</b>               | <a href="#">con_interval</a> | None for plots,<br>User prompt for grids. Batch: <b>0</b>                            | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.  |
| <b>-cb=base</b>                   | <a href="#">con_base</a>     | Lowest contour value, maximum wind speed   | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.  |
| <b>-fl=font_list</b>              | <a href="#">font_list</a>    | <b>modern.fnt</b>  | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>            | <a href="#">color_table</a>  | <b>wxp.clr</b>   | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>          | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse plots.                                      | Specifies the text color for labels above and below the plot.   |
| <b>-cod=color[:attr]</b>          | <a href="#">color_data</a>   | <b>green</b>   | Specifies the color for plotted data.   |
| <b>-com=color[:attr]</b>          | <a href="#">color_map</a>    | <b>red</b>   | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>         | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>  | Specifies the color for lat/lon lines.  |
| <b>-cowl=color[:attr]</b>         | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse plots.                                      | Specifies the color for wind barbs.   |
| <b>-cocd=color[:attr]</b>         | <a href="#">color_cloud</a>  | <b>white</b><br><b>black</b> for inverse plots.                                      | Specifies the color for cloud cover symbols.  |
| <b>-cowx=color[:attr]</b>         | <a href="#">color_wx</a>     | <b>yellow</b><br><b>brown</b> for inverse plots                                      | Specifies the color for weather symbols   |
| <b>-copp=color[:attr],</b><br>... | <a href="#">color_pop</a>    | <b>yellow</b><br><b>brown</b> for inverse plots                                      | Specifies the colors for probability of precipitation.  |

|  |                              |  |  |
|--|------------------------------|--|--|
| <b>-cocm=</b> <i>color[:attr]</i>          | <a href="#">color_cmap</a>   | <b>black</b>   | Specifies the color for color fill contour maps.   |
| <b>-coco=</b> <i>color[:attr]</i>          | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the color for line contours.   |
| <b>-cof=</b> <i>[val:]color[:attr],...</i> | <a href="#">color_fill</a>   | A predefined color set based on the <b>wxp.clr</b> color table | Specifies the color fill contour color set. See the <b>color_fill</b> resource for more details.                               |
| <b>-cola=</b> <i>color[:attr]</i>          | <a href="#">color_label</a>  | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the line contour label color.  |
| <b>-cocl=</b> <i>color[:attr]</i>          | <a href="#">color_clabel</a> | <b>black</b>   | Specifies the color fill contour label color.  |
| <b>-de=</b> <i>device[,...]</i>            | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                           | Specifies the output device.   |
| <b>-ti=</b> <i>title</i>                   | <a href="#">title</a>        | Program name   | Specifies the window title.  |
| <b>-ic</b>                                 | <a href="#">icon</a>         | <b>Off</b>   | Specifies whether to open window in iconified mode.  |
| <b>-bg=</b> <i>color</i>                   | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots                 | Specifies the background color of the window.  |
| <b>-ge=</b> <i>geometry</i>                | <a href="#">geometry</a>     | <b>640x480</b>   | Specifies the window and paper geometry.   |
| <i>filename[#seq]</i>                      | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: <b>current=la</b>                | The name of the MOS data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. |

## DESCRIPTION

This program contours MOS (Model Output Statistics) data on a background map. The input to the program is a raw ingested MOS data file. This is different from other WXP plotting programs in that MOS data is regularly formatted so a decode/conversion process is not necessary. Then, based on a plot domain, forecast time and a variable selected by the user, a plot is generated. Overlay plots, contour plots and loops can also be generated if the appropriate options are set.

The program will display data from 4 different model sources:

- **NGM** - MOS statistical forecast data derived from the NGM model. This includes many parameters such as max/min temperature, 3 hourly estimates of temperature, dewpoint, precipitation, winds, etc.
- **NGM-UA** - This is derived directly from the NGM model grids and represent data such as boundary layer temperature, relative humidity, sea level pressure, lifted index, 1000-500 mb thickness and vertical velocity.
- **ETA-UA** - This is very similar to the NGM-UA except the data are from the ETA model. This includes lowest level temperatures, relative humidity, sea level pressure, lifted index, 1000-500 mb thickness and vertical velocity.
- **MRFX** - This is derived extended range forecast data taken from the MRF model run once a day. The data include max/min temperature, probability of precipitation, winds, and cloud coverage reported once every 12 hours.

The model type can be changed with the [model](#) resource.

The plot domain defaults to the contiguous US unless changed through the [plot\\_domain](#) resource. The domain can be interactively changed using the mouse once the plot has been generated.

The programs starts off by prompting the user for input MOS file name. For some ingestors, such as the LDM, the list may contain many files that don't contain plottable MOS data. Only the 0 and 12Z data files are usable. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **mos\_dat** tag but this can be changed with the [in\\_file](#) resource.

**NOTE:** The program assumes that the model data for all model types are contained in a single file type represented by the **mos\_dat** tag. If the model types are in different files, the tag must be specified in the [in\\_file](#) resource. Both model and tag can be specified simultaneously if the model type is contained in the tag. For example, using **mos\_ngm\_ua** for the tag for NGM\_UA data. The program will see the "ngm\_ua" in the tag and use that model type.

Next, the program prompts the user for the forecast time. This is done through a forecast time menu which is specified in the **focalc.tim** file. Remember, each model type has a specific set of valid forecast times.

After the forecast time has been set, the city database is read in and a cross-reference table is setup. Since MOS data does not have location information, the program will refer to this table to location each station in the input file.

The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the **focalc.var** file. The internal variables are

#### Internal Variables (NGM)

| Variable                                   | Description and units                                       | Plots   |
|--|---|---|
| <b>city</b>                                | All stations in the data file                               | text  |
| <b>temp</b>                                | Temperature (F)   | value   |
| <b>dewp</b>                                | Dewpoint (F)  | value   |
| <b>dewd</b>                                | Dewpoint depression (F)                                     | value   |
| <b>wdir</b>                                | Wind direction (deg)  | value   |
| <b>wspd</b>                                | Wind speed (knt)  | value   |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)                              | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)                        | value   |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)                        | value   |
| <b>uwndg</b>                               | U wind component - grid relative (m/s)                      | value   |
| <b>vwndg</b>                               | V wind component - grid relative (m/s)                      | value   |
| <b>pop</b><br><b>pop6</b>                  | 6 hour probability of precipitation (%)                     | value   |
| <b>pop12</b>                               | 12 hour probability of precipitation (%)                    | value   |
| <b>pot</b><br><b>pot6</b>                  | 6 hour probability of thunderstorms (%)                     | value   |
| <b>pot12</b>                               | 12 hour probability of thunderstorms (%)                    | value   |
| <b>post</b><br><b>post6</b>                | 6 hour probability of severe thunderstorms (%)              | value   |
| <b>post12</b>                              | 12 hour probability of severe thunderstorms (%)             | value   |
| <b>pptype</b>                              | Precipitation type (number from MOS record)                 | value   |
| <b>obvis</b>                               | Obscuration to visibility (number from MOS record)          | value   |
| <b>wx</b><br><b>wx6</b>                    | Present weather based on 6 hour forecast data (SAO string)  | text  |
| <b>wx12</b>                                | Present weather based on 12 hour forecast data (SAO string) | text  |
| <b>vis</b>                                 | Visibility (mi)   | value   |
| <b>cldcl</b>                               | Cloud ceiling (100 ft)                                      | value   |
| <b>cldcv</b>                               | Cloud cover (cloud string)                                  | value   |
| <b>extt</b>                                | Extreme temperature (F) uses maximum or minimum             | value   |
| <b>maxt</b>                                | Maximum temperature (F) not always available                | value   |
| <b>mint</b>                                | Minimum temperature (F) not always available                | value   |
| <b>prec</b>                                | 6 hour precipitation total (in)                             | value   |

|                               |                                       |   |
|-------------------------------|---------------------------------------|---|
| <b>prec6</b>                  |                                       |   |
| <b>prec12</b>                 | 12 hour precipitation total (in)      | value                                     |
| <b>snwfl</b><br><b>snwfl6</b> | 6 hour snowfall total (in)            | value                                     |
| <b>snwfl12</b>                | 12 hour snowfall total (in)           | value                                     |
| <b>vpres</b>                  | Vapor pressure (mb)                   | value                                     |
| <b>wchilt</b>                 | Wind chill temperature (K)            | value                                     |
| <b>wchilf</b>                 | Wind chill factor (W/m <sup>2</sup> ) | value                                     |
| <b>heat</b>                   | Heat index (K)                        | value                                     |
| <b>humit</b>                  | Humiture (F)                          | value                                     |
| <b>rhum</b>                   | Relative humidity (%)                 | value                                     |
| <b>fr</b>                     | Flight rules                          | value, text<br>0=VFR,<br>1=MVFR,<br>2=IFR |

#### Internal Variables (MRF)

| Variable                   | Description and units   | Plots |
|----------------------------|---|-------|
| <b>city</b>                | All stations in the data file   | text  |
| <b>temp</b><br><b>maxt</b> | Maximum temperature (F)   | value |
| <b>nmaxt</b>               | Normal maximum temperature (F)  | value |
| <b>mint</b>                | Minimum temperature (F)   | value |
| <b>dmint</b>               | Normal minimum temperature (F)  | value |
| <b>wspd</b>                | Wind speed (knt) 00 and 12 Z value (specified with time resource)                         | value |
| <b>pop12</b>               | 12 hour probability of precipitation (%) 00 and 12 Z value (specified with time resource) | value |
| <b>pop</b><br><b>pop24</b> | 24 hour probability of precipitation (%)  | value |
| <b>pos</b>                 | 12 hour probability of snow (%) 00 and 12 Z value (specified with time resource)          | value |
| <b>wx</b>                  | Present weather (SAO string)  | text  |
| <b>cldev</b>               | Cloud cover (%) 00 and 12 Z value (specified with time resource)                          | value |

#### Internal Variables (UA)

| Variable                                   | Description and units  | Plots |
|--|--|-------|
| <b>city</b>                                | All stations in the data file                                  | text  |
| <b>temp</b>                                | Lowest level temperature (C)                                   | value |
| <b>tempb</b>                               | Boundary layer temp (C)  | value |
| <b>temp1</b>                               | Level 1 temperature (C)  | value |
| <b>temp2</b>                               | Level 2 temperature (C)  | value |
| <b>temp3</b>                               | Level 3 temperature (C)  | value |
| <b>wdir</b>                                | Wind direction (deg)   | value |
| <b>wspd</b>                                | Wind speed (knt)   | value |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)                                 | value |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)                           | value |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)                           | value |
| <b>uwndg</b>                               | U wind component - grid relative (m/s)                         | value |
| <b>vwndg</b>                               | V wind component - grid relative (m/s)                         | value |
| <b>pres</b>                                | Sea level pressure (mb)  | value |
| <b>prec</b>                                | Precipitation (in) use time variable to get accumulated totals | value |

|              |   |       |
|--------------|---|-------|
| <b>cprec</b> | 48 hour cumulative precipitation (in)     | value |
| <b>rhum</b>  | Lowest level relative humidity (%)        | value |
| <b>mrhum</b> | Mean boundary layer relative humidity (%) | value |
| <b>rhum1</b> | Level 1 relative humidity (%)             | value |
| <b>rhum2</b> | Level 2 relative humidity (%)             | value |
| <b>rhum3</b> | Level 3 relative humidity (%)             | value |
| <b>vvel</b>  | 700 mb vertical velocity (-ub/s)          | value |
| <b>lift</b>  | Lifted index (dC)                         | value |
| <b>thick</b> | 1000-500 mb thickness (m)                 | value |

These parameters can be accessed by name if they have not been redefined in the **focalc.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

There are several derived variables from this list which are defined in the **focalc.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable      | Description   | Plots  |
|---------------|---|--------|
| <b>all</b>    | All data, a composite plot of temperature (UL), dewpoint (LL), 6 hour probability of precipitation (yellow-UR), cloud cover, winds and present weather. | comp   |
| <b>cli</b>    | Climatic data, a composite plot of extreme temperature (UL), 12 hour POP (yellow-UR), cloud cover, winds and present weather.                           | comp   |
| <b>allm</b>   | All data for MRF, a composite plot of max (UL) and min (LL) temperature, 24 hour POP (yellow-UR), cloud cover, present weather                          | comp   |
| <b>depict</b> | Weather depiction plot including flight rules (contoured), cloud cover, cloud ceiling and present weather.  | comp   |
| <b>wind</b>   | A vector quantity of <uwndg,vwndg>  | vector |
| <b>conv</b>   | Wind convergence (/s) positive is convergence   | value  |
| <b>mconv</b>  | Moisture convergence (/s)   | value  |
| <b>rvort</b>  | Relative vorticity (/s)   | value  |
| <b>avort</b>  | Absolute vorticity (/s)   | value  |
| <b>tadv</b>   | Temperature advection (K/s)   | value  |
| <b>madv</b>   | Moisture advection (/s)   | value  |
| <b>vadv</b>   | Vorticity advection (/s^2)  | value  |

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the data file is opened and data read in. Any additional computations are performed to produce a derived variable and fit the data to a grid. The background map will plot first unless the output is a color fill contour in which the map plots last. The gridding process uses an objective analysis scheme to produce the grids. The parameters for this process are specified in the **object\_param** resource. Once the grids have been created, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.
- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

## Present Weather Determination

The estimated type of weather on NGM plots is based on the following information:

- Precipitation based on probability of precipitation > 30%,
- Precipitation type based on PTYPE field (rain, snow, freezing rain),
- Precipitation intensity based on quantitative precipitation QPF field (see below),
- Existence of thunderstorms based on probability of thunderstorm > 20%,
- Existence of severe thunderstorms based on severe thunderstorm probability > 30%,
- Existence of obscuration (fog/haze) based on VIS field < 1 mile where obscuration type determined by OBVIS field.

### Precipitation type and intensity based on 6 hour information

| Type     | 1   | 2   | 3  | 4   | 5   |
|----------|-----|-----|----|-----|-----|
| Rain     | R-  | R-  | R- | R   | R+  |
| Freezing | ZR- | ZR- | ZR | ZR+ | ZR+ |
| Snow     | S-  | S-  | S  | S+  | S+  |

### Precipitation type and intensity based on 12 hour information

| Type     | 1   | 2   | 3   | 4  | 5   | 6   |
|----------|-----|-----|-----|----|-----|-----|
| Rain     | R-  | R-  | R-  | R- | R   | R+  |
| Freezing | ZR- | ZR- | ZR- | ZR | ZR+ | ZR+ |
| Snow     | S-  | S-  | S-  | S  | S+  | S+  |

## EXAMPLES

```
focalc -cu=la -ft=h24 -va=extt -de=d -pl=cf -in=2
```

This contours the extreme temperature for the US in color fills with an interval of 2.

```
focalc -cu=la -ft=h24 -va=temp -of=wxp -pl=none
```

This will generate grid file based on the **mos\_grd** name convention. A specific filename can be used instead of **wxp**.

```
focalc -mo=mrfx -cu=la -ft=d4 -va=maxt
```

This contours maximum temperatures from the MRF MOS.

## FILES

- **mos.cty** - the surface station location database file
- **focalc.tim** - the time menu
- **focalc.var** - the variable menu and definitions

## SEE ALSO

- [fouswx](#) - the MOS data plotting program
- [statlog](#) - the MOS meteogram plotting program

# FORECAST

## NAME

**forecast** - Forecast data parsing program

## SYNOPSIS

**forecast** [*parameters...*] *filename*

## PARAMETERS

| Command Line                         | Resource                  | Default             | Description  |
|--------------------------------------|---------------------------|---------------------|--|
| <b>-h</b>                            | <a href="#">help</a>      | No                  | Lists basic help information.  |
| <b>-df=<i>filename</i></b>           | <a href="#">default</a>   | .wxpdef             | Sets the name of the resource file.  |
| <b>-na=<i>name</i></b>               | <a href="#">name</a>      | forecast            | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                           | <a href="#">batch</a>     | No                  | Run program in batch mode  |
| <b>-me=<i>level</i></b>              | <a href="#">message</a>   | out2                | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>   |
| <b>-fp=<i>filepath</i></b>           | <a href="#">file_path</a> | current directory   | Specifies location of database files.  |
| <b>-dp=<i>datapath</i></b>           | <a href="#">data_path</a> | current directory   | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-nc=<i>name_conv</i></b>          | <a href="#">name_conv</a> | name_conv           | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=<i>in_file</i></b>            | <a href="#">in_file</a>   | for_dat             | Specifies the input file name tag. The default tag is <b>for_dat</b> but may need to be changed for specific forecast types.   |
| <b>-cu=[<i>hour</i> <i>la</i>]</b>   | <a href="#">current</a>   | None                | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <i>la</i> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=<i>hour</i></b>               | <a href="#">hour</a>      | None                | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=<i>num_hour</i></b>           | <a href="#">num_hour</a>  | 0                   | This specifies the number of hours that will be used. If this is not specified, a single hour will be parse. Otherwise a set of hours will be parsed.  |
| <b>-fc=<i>forecast</i></b>           | <a href="#">forecast</a>  | User Prompt         | This specifies the product header to search for. <ul style="list-style-type: none"> <li><b>state</b> -- Prints the state forecast (FPUS1)</li> <li><b>disc</b> -- Prints the state forecast discussions (FPUS3)</li> <li><b>cod_city</b> -- Prints the coded city forecasts (FPUS4)</li> <li><b>zones</b> -- Prints the zone forecasts (FPUS(05 50 51 52 53 54 55 56 61 62 71 72))</li> <li><b>zone</b> -- Prints a specific zone forecast. The user is prompted for the zone.</li> <li><b>%zone</b> -- Prints a specific zone forecast</li> </ul> |
| <b>-st=<i>state</i></b>              | <a href="#">state</a>     | None                | Specifies the state weather forecast office to use. This is a 4 letter ID.   |
| <b>-pa=<i>param</i></b>              | <a href="#">parameter</a> | None                | Specifies additional plotting parameters.  |
| <b><i>filename</i> [#<i>seq</i>]</b> | <a href="#">filename</a>  | None<br>User Prompt | The name of the surface data file to be used. An optional sequence number can be added to designate the time for non-Batch: current= <b>la</b> WXP files.  |

## DESCRIPTION

This program parses forecast data for a specific forecast product from a specific WFO. The input to the program is a ingested forecast data file. The name of the input file is determined through the [in file](#) resource. The default tag is **for\_dat**.

The program starts off by prompting the user for input data file name. The user may specify the input file either via the command line or through the [current](#) resource.

Next, a menu of possible forecasts is listed and the user enters the forecast type. If the forecast type is a zone forecast, the user will also be prompted for a zone. A sample zone would be "INZ029".

### Header Files

The use of a header file can considerably improve access to data files. Rather than parsing the entire file which at times is larger than 1MB, the product headers can be parsed directly out of a header file. Header files are much smaller and parse very fast. The header file contains a byte offset into the large file.

## EXAMPLES

To parse for the latest state forecast from KIND

```
forecast -cu -nh=-12 -fc=state -st=KIND
** FPUS1 KIND 022030 ***
SFPIN
INZ002>089-031000-
```

```
STATE FORECAST FOR INDIANA
NATIONAL WEATHER SERVICE INDIANAPOLIS IN
330 PM EST THU OCT 2 1997
```

```
.TONIGHT...FAIR AND WARMER. LOWS 50 TO 55.
.FRIDAY...MOSTLY SUNNY...BREEZY AND WARMER. HIGHS 80 TO 85.
.FRIDAY NIGHT...BECOMING MOSTLY CLOUDY. A CHANCE OF THUNDERSTORMS.
LOWS IN THE LOWER 60S.
.SATURDAY...MOSTLY CLOUDY...BREEZY AND A CHANCE OF THUNDERSTORMS.
WARM. HIGHS MIDDLE 70S TO AROUND 80.
```

```
.EXTENDED FORECAST...
.SUNDAY AND MONDAY...MOSTLY CLEAR AND WARM. LOWS MIDDLE 50S TO AROUND
60. HIGHS UPPER 70S TO LOWER 80S.
.TUESDAY...PARTLY CLOUDY AND MILD. LOWS AROUND 50 TO MIDDLE 50S.
HIGHS IN THE 70S.
DS
```

To parse for the latest zone forecast

```
forecast -cu -nh=-12 -fc=%INZ029 -st=KIND
** FPUS53 KIND 022040 COR ***
INZ020>023-028>030-035-036-043-044-051-052-060-067-030930-
CARROLL-CASS-CLAY-CLINTON-FOUNTAIN-KNOX-MIAMI-MONTGOMERY-PARKE-
SULLIVAN-TIPPECANOE-VERMILLION-VIGO-WARREN-WHITE-
INCLUDING THE CITIES OF...CRAWFORDSVILLE...FRANKFORT...LAFAYETTE...
LOGANSPO...TERRE HAUTE...VINCENNES
330 PM EST THU OCT 2 1997
```

```
.TONIGHT...PARTLY CLOUDY AND WARMER. LOW IN THE MIDDLE 50S. SOUTHWEST
WIND 5 TO 10 MPH.
```

.FRIDAY...MOSTLY SUNNY AND WARMER. HIGH 80 TO 85. BREEZY SOUTHWEST WIND 15 TO 20 MPH.

.FRIDAY NIGHT...BECOMING MOSTLY CLOUDY. A 40 PERCENT CHANCE OF THUNDERSTORMS. MILD. LOW IN THE LOWER 60S.

.SATURDAY...MOSTLY CLOUDY...BREEZY AND A 40 PERCENT CHANCE OF THUNDERSTORMS. MILD. HIGH IN THE UPPER 70S.

## FILES

## SEE ALSO

- [parse](#) - the text data parsing program

# FOUSWX

## NAME

**fouswx** - MOS data plotting program

## SYNOPSIS

**fouswx** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description  |
|------------------------------|----------------------------|-------------------|--|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.  |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxpdef           | Sets the name of the resource file.  |
| <b>-na=name</b>              | <a href="#">name</a>       | fouswx            | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode  |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• gridding information - <b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | mos_dat           | Specifies the input file name tag. The default is <b>mos_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of raw or grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII grid or raw file</li> <li>• <b>wxpb</b> - WXP binary grid file</li> <li>• <b>raw</b> - WXP ASCII raw file</li> <li>• <b>cdf</b> - netCDF binary grid file</li> </ul> Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>mos_raw</b> or <b>mos_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type: <code>wxp:mos_grd</code><br>A full filename can be specified which can contain name convention tags. |
| <b>-pf=file_param</b>        | <a href="#">file_param</a> | User prompt       | This specifies how to treat output files that exist:   |

|   |                              |  |  |
|---|------------------------------|--|--|
|   |                              | Batch: <b>use</b>  | <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul>             |
| <b>-cf=cityfile</b>   | <a href="#">city file</a>    | <b>mos.cty</b>   | Specifies the name of the city database file used in finding station locations.  |
| <b>-mf=mapfile[,...]</b>  | <a href="#">map file</a>     | <b>wxp.map</b> or <b>cont.dat</b> , <b>country.dat</b> for domains outside North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu=[hour la]</b>  | <a href="#">current</a>      | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=hour</b>   | <a href="#">hour</a>         | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-mo=model</b>  | <a href="#">model</a>        | <b>ngm</b>   | Specifies the model to use as input data. <ul style="list-style-type: none"> <li>• <b>ngm</b> -- Use Nested Grid Model (NGM) MOS data (FOUS14) (DEFAULT),</li> <li>• <b>mrfx</b> -- Use Medium Range Forecast (MRF) MOS data (FOX),</li> <li>• <b>ngm_ua</b> -- Use upper air NGM MOS (FOUE,M,W60-78)</li> <li>• <b>eta_ua</b> -- Use upper air ETA MOS (FOUS60-78)</li> </ul> |
| <b>-id=identifier</b>   | <a href="#">identifier</a>   | None   | Used to plot a specific observation. If not specified, all stations will be plotted.   |
| <b>-pd=domain</b>   | <a href="#">plot domain</a>  | User prompt<br>Batch: first domain in <b>wxp.reg</b> file.                               | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-gd=grid_domain</b>  | <a href="#">grid domain</a>  | Same as plot domain  | This specifies the domain to use when fitting data to a grid. If this is not specified, it is the <b>plot_domain</b> .   |
| <b>-re=region</b>   | <a href="#">region</a>       | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-pr=prior</b>  | <a href="#">stat prior</a>   | Depends on domain size   | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.   |
| <b>-ft=time</b>   | <a href="#">time</a>         | User prompt<br>Batch: initial time   | This specifies the forecast time to be used in the plots. To create loops, a list of forecast time are specified separated by commas.  |
| <b>-va=variable</b>   | <a href="#">variable</a>     | User prompt<br>Batch: first variable listed in the <b>sfcwx.var</b> file                 | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.   |
| <b>-oa=smooth</b><br>[,radinf][,passes]<br>[,converg]<br>[,min_sta] | <a href="#">object param</a> | <b>1,3,1,,2,1</b>  | Specifies the objective analysis parameters. There are 5 parameters including smoothing, radius of influence, number of passes, convergence parameter and minimum number of stations.  |
| <b>-pl=plot_type</b>  | <a href="#">plot type</a>    | <b>Data Plot</b><br>User prompt if   | Specifies the type of output plot. By default, this is a simple data plot. To produce a contour or vector plot, use the  |

|                                   |                              |  |   |
|-----------------------------------|------------------------------|--|---|
|                                   |                              | plotting grids<br>Batch: fill<br>contours or<br>vectors      | following plot types:<br><b>Contours:</b> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <b>Vectors:</b> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>           | <a href="#">plot_scale</a>   | <b>1.0</b>   | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>                 | <a href="#">plot_format</a>  | None   | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.  |
| <b>-pp=param</b><br>[,param...]   | <a href="#">plot_param</a>   | None   | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>         | <a href="#">draw</a>         | <b>all</b>   | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>            | <a href="#">label</a>        | Program defaults   | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.   |
| <b>-in=interval</b>               | <a href="#">con_interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b> | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.  |
| <b>-cb=base</b>                   | <a href="#">con_base</a>     | Lowest contour<br>value, maximum<br>wind speed               | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.  |
| <b>-fl=font_list</b>              | <a href="#">font_list</a>    | <b>modern.fnt</b>  | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>            | <a href="#">color_table</a>  | <b>wxp.clr</b>   | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>          | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.           | Specifies the text color for labels above and below the plot.   |
| <b>-cod=color[:attr]</b>          | <a href="#">color_data</a>   | <b>green</b>   | Specifies the color for plotted data.   |
| <b>-com=color[:attr]</b>          | <a href="#">color_map</a>    | <b>red</b>   | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>         | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>  | Specifies the color for lat/lon lines.  |
| <b>-cowl=color[:attr]</b>         | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.           | Specifies the color for wind barbs.   |
| <b>-codd=color[:attr]</b>         | <a href="#">color_cloud</a>  | <b>white</b><br><b>black</b> for inverse<br>plots.           | Specifies the color for cloud cover symbols.  |
| <b>-cowx=color[:attr]</b>         | <a href="#">color_wx</a>     | <b>yellow</b><br><b>brown</b> for<br>inverse plots           | Specifies the color for weather symbols   |
| <b>-copp=color[:attr],</b><br>... | <a href="#">color_pop</a>    | <b>yellow</b><br><b>brown</b> for<br>inverse plots           | Specifies the colors for probability of precipitation.  |
| <b>-cocm=color[:attr]</b>         | <a href="#">color_cmap</a>   | <b>black</b>   | Specifies the color for color fill contour maps.  |
| <b>-coco=color[:attr]</b>         | <a href="#">color_cont</a>   | <b>white</b>   | Specifies the color for line contours.  |

|                                     |                              |  |  |
|-------------------------------------|------------------------------|--|--|
|                                     |                              | <b>black</b> for inverse plots.                                |  |
| <b>-cof</b> =[val:]color[:attr],... | <a href="#">color_fill</a>   | A predefined color set based on the <b>wxp.clr</b> color table | Specifies the color fill contour color set. See the <b>color_fill</b> resource for more details.                               |
| <b>-cola</b> =color[:attr]          | <a href="#">color_label</a>  | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the line contour label color.  |
| <b>-cocl</b> =color[:attr]          | <a href="#">color_clabel</a> | <b>black</b>   | Specifies the color fill contour label color.  |
| <b>-de</b> =device[,...]            | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                           | Specifies the output device.   |
| <b>-ti</b> =title                   | <a href="#">title</a>        | Program name   | Specifies the window title.  |
| <b>-ic</b>                          | <a href="#">icon</a>         | <b>Off</b>   | Specifies whether to open window in iconified mode.  |
| <b>-bg</b> =color                   | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots                 | Specifies the background color of the window.  |
| <b>-ge</b> =geometry                | <a href="#">geometry</a>     | <b>640x480</b>   | Specifies the window and paper geometry.   |
| filename[#seq]                      | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: <b>current=la</b>                | The name of the MOS data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. |

## DESCRIPTION

This program plots MOS (Model Output Statistics) data on a background map. The input to the program is a raw ingested MOS data file. This is different from other WXP plotting programs in that MOS data is regularly formatted so a decode/conversion process is not necessary. Then, based on a plot domain, forecast time and a variable selected by the user, a plot is generated. Overlay plots, contour plots and loops can also be generated if the appropriate options are set.

The program will display data from 4 different model sources:

- **NGM** - MOS statistical forecast data derived from the NGM model. This includes many parameters such as max/min temperature, 3 hourly estimates of temperature, dewpoint, precipitation, winds, etc.
- **NGM-UA** - This is derived directly from the NGM model grids and represent data such as boundary layer temperature, relative humidity, sea level pressure, lifted index, 1000-500 mb thickness and vertical velocity.
- **ETA-UA** - This is very similar to the NGM-UA except the data are from the ETA model. This includes lowest level temperatures, relative humidity, sea level pressure, lifted index, 1000-500 mb thickness and vertical velocity.
- **MRFX** - This is derived extended range forecast data taken from the MRF model run once a day. The data include max/min temperature, probability of precipitation, winds, and cloud coverage reported once every 12 hours.

The model type can be changed with the [model](#) resource.

The plot domain defaults to the contiguous US unless changed through the [plot\\_domain](#) resource. The domain can be interactively changed using the mouse once the plot has been generated.

The programs starts off by prompting the user for input MOS file name. For some ingestors, such as the LDM, the list may contain many files that don't contain plottable MOS data. Only the 0 and 12Z data files are usable. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **mos\_dat** tag but this can be changed with the [in\\_file](#) resource.

**NOTE:** The program assumes that the model data for all model types are contained in a single file type represented by the **mos\_dat** tag. If the model types are in different files, the tag must be specified in the **in\_file** resource. Both model and tag can be specified simultaneously if the model type is contained in the tag. For example, using **mos\_ngm\_ua** for the tag for NGM\_UA data. The program will see the "ngm\_ua" in the tag and use that model type.

Next, the program prompts the user for the forecast time. This is done through a forecast time menu which is specified in the **fouswx.tim** file. Remember, each model type has a specific set of valid forecast times.

After the forecast time has been set, the city database is read in and a cross-reference table is setup. Since MOS data does not have location information, the program will refer to this table to location each station in the input file. The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the **fouswx.var** file. The internal variables are

#### Internal Variables (NGM)

| Variable                                   | Description and units                                       | Plots   |
|--|---|---|
| <b>city</b>                                | All stations in the data file                               | text  |
| <b>temp</b>                                | Temperature (F)   | value   |
| <b>dewp</b>                                | Dewpoint (F)  | value   |
| <b>dewd</b>                                | Dewpoint depression (F)                                     | value   |
| <b>wdir</b>                                | Wind direction (deg)  | value   |
| <b>wspd</b>                                | Wind speed (knt)  | value   |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)                              | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)                        | value   |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)                        | value   |
| <b>uwndg</b>                               | U wind component - grid relative (m/s)                      | value   |
| <b>vwndg</b>                               | V wind component - grid relative (m/s)                      | value   |
| <b>pop</b><br><b>pop6</b>                  | 6 hour probability of precipitation (%)                     | value   |
| <b>pop12</b>                               | 12 hour probability of precipitation (%)                    | value   |
| <b>Pot</b><br><b>pot6</b>                  | 6 hour probability of thunderstorms (%)                     | value   |
| <b>pot12</b>                               | 12 hour probability of thunderstorms (%)                    | value   |
| <b>post</b><br><b>post6</b>                | 6 hour probability of severe thunderstorms (%)              | value   |
| <b>post12</b>                              | 12 hour probability of severe thunderstorms (%)             | value   |
| <b>pptype</b>                              | Precipitation type (number from MOS record)                 | value   |
| <b>obvis</b>                               | Obscuration to visibility (number from MOS record)          | value   |
| <b>wx</b><br><b>wx6</b>                    | Present weather based on 6 hour forecast data (SAO string)  | text  |
| <b>wx12</b>                                | Present weather based on 12 hour forecast data (SAO string) | text  |
| <b>vis</b>                                 | Visibility (mi)   | value   |
| <b>cldcl</b>                               | Cloud ceiling (100 ft)                                      | value   |
| <b>cldcv</b>                               | Cloud cover (cloud string)                                  | value   |
| <b>extt</b>                                | Extreme temperature (F) uses maximum or minimum             | value   |
| <b>maxt</b>                                | Maximum temperature (F) not always available                | value   |
| <b>mint</b>                                | Minimum temperature (F) not always available                | value   |
| <b>prec</b><br><b>prec6</b>                | 6 hour precipitation total (in)                             | value   |
| <b>prec12</b>                              | 12 hour precipitation total (in)                            | value   |

|                               |  |   |
|-------------------------------|--|---|
| <b>snwfl</b><br><b>snwfl6</b> | 6 hour snowfall total (in)                             | value                                     |
| <b>snwfl12</b>                | 12 hour snowfall total (in)                            | value                                     |
| <b>vpres</b>                  | Vapor pressure (mb)                                    | value                                     |
| <b>wchilt</b>                 | Wind chill temperature (K)                             | value                                     |
| <b>wchilf</b>                 | Wind chill factor (W/m <sup>2</sup> )                  | value                                     |
| <b>heat</b>                   | Heat index (K)   | value                                     |
| <b>humit</b>                  | Humiture (F)   | value                                     |
| <b>rhum</b>                   | Relative humidity (%)                                  | value                                     |
| <b>fr</b>                     | Flight rules (number and string: 0=VFR, 1=MVFR, 2=IFR) | value, text<br>0=VFR,<br>1=MVFR,<br>2=IFR |

#### Internal Variables (MRF)

| Variable                   | Description and units   | Plots |
|----------------------------|---|-------|
| <b>city</b>                | All stations in the data file   | text  |
| <b>temp</b><br><b>maxt</b> | Maximum temperature (F)   | value |
| <b>nmact</b>               | Normal maximum temperature (F)  | value |
| <b>mint</b>                | Minimum temperature (F)   | value |
| <b>dmint</b>               | Normal minimum temperature (F)  | value |
| <b>wspd</b>                | Wind speed (knt) 00 and 12 Z value (specified with time resource)                         | value |
| <b>pop12</b>               | 12 hour probability of precipitation (%) 00 and 12 Z value (specified with time resource) | value |
| <b>pop</b><br><b>pop24</b> | 24 hour probability of precipitation (%)  | value |
| <b>pos</b>                 | 12 hour probability of snow (%) 00 and 12 Z value (specified with time resource)          | value |
| <b>wx</b>                  | Present weather (SAO string)  | text  |
| <b>cldcv</b>               | Cloud cover (%) 00 and 12 Z value (specified with time resource)                          | value |

#### Internal Variables (UA)

| Variable                                   | Description and units  | Plots |
|--|--|-------|
| <b>city</b>                                | All stations in the data file                                  | text  |
| <b>temp</b>                                | Lowest level temperature (C)                                   | value |
| <b>tempb</b>                               | Boundary layer temp (C)  | value |
| <b>temp1</b>                               | Level 1 temperature (C)  | value |
| <b>temp2</b>                               | Level 2 temperature (C)  | value |
| <b>temp3</b>                               | Level 3 temperature (C)  | value |
| <b>wdir</b>                                | Wind direction (deg)   | value |
| <b>wspd</b>                                | Wind speed (knt)   | value |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)                                 | value |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)                           | value |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)                           | value |
| <b>uwndg</b>                               | U wind component - grid relative (m/s)                         | value |
| <b>vwndg</b>                               | V wind component - grid relative (m/s)                         | value |
| <b>pres</b>                                | Sea level pressure (mb)  | value |
| <b>prec</b>                                | Precipitation (in) use time variable to get accumulated totals | value |
| <b>cprec</b>                               | 48 hour cumulative precipitation (in)                          | value |
| <b>rhum</b>                                | Lowest level relative humidity (%)                             | value |

|              |   |       |
|--------------|---|-------|
| <b>mrhum</b> | Mean boundary layer relative humidity (%) | value |
| <b>rhum1</b> | Level 1 relative humidity (%)             | value |
| <b>rhum2</b> | Level 2 relative humidity (%)             | value |
| <b>rhum3</b> | Level 3 relative humidity (%)             | value |
| <b>vvel</b>  | 700 mb vertical velocity (-ub/s)          | value |
| <b>lift</b>  | Lifted index (dC)                         | value |
| <b>thick</b> | 1000-500 mb thickness (m)                 | value |

These parameters can be accessed by name if they have not been redefined in the **fouswx.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

There are several derived variables from this list which are defined in the **fouswx.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable      | Description   | Plots  |
|---------------|---|--------|
| <b>all</b>    | All data, a composite plot of temperature (UL), dewpoint (LL), 6 hour probability of precipitation (yellow-UR), cloud cover, winds and present weather. | comp   |
| <b>cli</b>    | Climatic data, a composite plot of extreme temperature (UL), 12 hour POP (yellow-UR), cloud cover, winds and present weather.                           | comp   |
| <b>allm</b>   | All data for MRF, a composite plot of max (UL) and min (LL) temperature, 24 hour POP (yellow-UR), cloud cover, present weather                          | comp   |
| <b>depict</b> | Weather depiction plot including flight rules (contoured), cloud cover, cloud ceiling and present weather.  | comp   |
| <b>wind</b>   | A vector quantity of <uwndg,vwndg>  | vector |
| <b>conv</b>   | Wind convergence (/s) positive is convergence   | value  |
| <b>mconv</b>  | Moisture convergence (/s)   | value  |
| <b>rvort</b>  | Relative vorticity (/s)   | value  |
| <b>avort</b>  | Absolute vorticity (/s)   | value  |
| <b>tadv</b>   | Temperature advection (K/s)   | value  |
| <b>madv</b>   | Moisture advection (/s)   | value  |
| <b>vadv</b>   | Vorticity advection (/s^2)  | value  |

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the data file is opened and data read in. Any additional computations are performed and the data are plotted. The background map will plot first unless the output is a color fill contour in which the map plots last. If a simple plot is specified such as "**temp**", the data will be plotted to 0 decimal places on the map. Formatting of the number can be done with the **plot\_format** resource. Composite plots can either be specified with the variable resource (.var file, resource file, command line or variable prompt). The variable resource allows the user to control much of the data formatting.

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed.

When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

#### Contour/Vector Plots

By default, this program plots data but it can produce grids that can either be contoured or plotted as vectors or streamlines. By specifying the **plot\_type** resource to a contour or vector type, the program will generate gridded fields rather than plotting the data. The gridding process uses an objective analysis scheme to produce the grids. The parameters for this process are specified in the **object\_param** resource. Once the grids have been created, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display in much the same fashion as for the data plot.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.
- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

## Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

## Present Weather Determination

The estimated type of weather on NGM plots is based on the following information:

- Precipitation based on probability of precipitation > 30%,
- Precipitation type based on PTYPE field (rain, snow, freezing rain),
- Precipitation intensity based on quantitative precipitation QPF field (see below),
- Existence of thunderstorms based on probability of thunderstorm > 20%,
- Existence of severe thunderstorms based on severe thunderstorm probability > 30%,
- Existence of obscuration (fog/haze) based on VIS field < 1 mile where obscuration type determined by OBVIS field.

### Precipitation type and intensity based on 6 hour information

| Type     | 1   | 2   | 3  | 4   | 5   |
|----------|-----|-----|----|-----|-----|
| Rain     | R-  | R-  | R- | R   | R+  |
| Freezing | ZR- | ZR- | ZR | ZR+ | ZR+ |
| Snow     | S-  | S-  | S  | S+  | S+  |

### Precipitation type and intensity based on 12 hour information

| Type     | 1   | 2   | 3   | 4  | 5   | 6   |
|----------|-----|-----|-----|----|-----|-----|
| Rain     | R-  | R-  | R-  | R- | R   | R+  |
| Freezing | ZR- | ZR- | ZR- | ZR | ZR+ | ZR+ |
| Snow     | S-  | S-  | S-  | S  | S+  | S+  |

## EXAMPLES

```
fouswx -cu=la -ft=h24 -va=all -de=d
```

This plots a surface composite plot containing temperature, dewpoint, probability of precipitation, clouds, winds and weather for the US.

```
fouswx -cu=la -ft=h24 -va=temp -of=raw -pl=none
```

This will generate raw file based on the **mos\_raw** name convention. A specific filename can be used instead of **raw**.

```
fouswx -cu=la -ft=h24 -va=temp -pl=cf -in=2 -de=d
```

This will produce a contour plot of temperature. Without the "-pl=cf", the program will just plot the temperature data.

```
fouswx -mo=mrfx -cu=la -ft=d4 -va=maxt
```

This plots maximum temperatures from the MRF MOS.

```
fouswx -mo=mrfx -cu=la -ft=e4 -va=pop12
```

This plots the 4.5 day 12 hour POP. This will be valid at 12Z.

## FILES

- **mos.cty** - the surface station location database file
- **fouswx.tim** - the time menu
- **fouswx.var** - the variable menu and definitions

## SEE ALSO

- [focalc](#) - the MOS data contouring program
- [statlog](#) - the MOS meteogram plotting program

# GIFCAT

## NAME

**gifcat** - GIF concatenation program

## SYNOPSIS

**gifcat** [*options...*] [*file...*]

## PARAMETERS

| Options                    | Description  |
|----------------------------|--|
| <b>-h</b>                  | Lists help information.  |
| <b>-s</b>                  | Silent mode, no information printed  |
| <b>-d</b> <i>delay</i>     | Specifies the delay between each image in .01 seconds. (Default 20)            |
| <b>-e</b> <i>end_delay</i> | Specifies the delay of the last image in the loop in .01 seconds. (Default 80) |
| <b>-o</b> <i>file</i>      | Specifies output GIF file.   |
| <i>file...</i>             | A list of GIF files to animate   |

## DESCRIPTION

This program takes each of the GIF files strips the trailer and adds the GIF89 extension block to link images together in a loop. Any number of files can be integrated into the loop. Loop delays can be specified. The output is a GIF file ready for display on the web.

```
gifcat -s -o eta_pres_loop.gif \
  eta_pres_init.gif eta_pres_6h.gif eta_pres_12h.gif eta_pres_18h.gif \
  eta_pres_24h.gif eta_pres_30h.gif eta_pres_36h.gif eta_pres_42h.gif \
  eta_pres_48h.gif
```

## SEE ALSO

- [pscat](#) - Postscript concatenation program
- [hpglcat](#) - HPGL concatenation program

# GRBCALC

## NAME

**grbcalc** - GRIB model contouring program

## SYNOPSIS

**grbcalc** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description   |
|------------------------------|----------------------------|-------------------|---|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.   |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wpxdef           | Sets the name of the resource file.   |
| <b>-na=name</b>              | <a href="#">name</a>       | grbcalc           | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode   |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>  |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.   |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input converted data files. This may be modified in the name convention file.  |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.   |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.   |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.   |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | grib              | Specifies the input file name tag. This value can also be specified with the <b>model</b> resource by looking up the naming convention in the <b>model.lup</b> file. The default is <b>grib</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII grid or raw file</li> <li><b>wxpb</b> - WXP binary grid file</li> <li><b>raw</b> - WXP ASCII raw file</li> <li><b>cdf</b> - netCDF binary grid file</li> </ul> <p>Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>grd_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type:<br/>wpx:eta_grd</p> |

|   |                             |  |   |
|---|-----------------------------|--|---|
|   |                             |  | A full filename can be specified which can contain name convention tags.  |
| <b>-pf=</b> <i>file_param</i>           | <a href="#">file_param</a>  | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul>                                     |
| <b>-mf=</b> <i>mapfile</i> [,...]       | <a href="#">map_file</a>    | <b>wxp.map</b> or <b>cont.dat</b> , <b>country.dat</b> for domains outside North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ] | <a href="#">current</a>     | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=</b> <i>hour</i>                 | <a href="#">hour</a>        | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-mo=</b> <i>model</i>                | <a href="#">model</a>       | <b>ngm</b> or first model listed in <b>model.lup</b> file.                               | Specifies the model to use as input data. These are defined in the <b>model.lup</b> file. This file can define how model grids are pieced together. Some samples are: <ul style="list-style-type: none"> <li>• <b>ngm</b> -- Nested Grid Model</li> <li>• <b>eta</b> -- ETA model</li> <li>• <b>avn</b> -- Aviation model</li> <li>• <b>mrf</b> -- Medium Range Forecast model</li> </ul>   |
| <b>-pd=</b> <i>domain</i>               | <a href="#">plot_domain</a> | User prompt<br>Batch: first domain in <b>wxp.reg</b> file.                               | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-re=</b> <i>region</i>               | <a href="#">region</a>      | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-ft=</b> <i>time</i>                 | <a href="#">time</a>        | User prompt<br>Batch: initial time   | This specifies the forecast time to be used in the plots. To create loops, a list of forecast time are specified separated by commas.   |
| <b>-le=</b> <i>level</i>                | <a href="#">level</a>       | User Prompt<br>Batch: <b>500</b>   | This specifies the vertical level to use in the plot. See the <b>level</b> resource for more information on what can be specified.  |
| <b>-va=</b> <i>variable</i>             | <a href="#">variable</a>    | User prompt<br>Batch: first variable listed in the <b>grbcalc.var</b> file               | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.  |
| <b>-pl=</b> <i>plot_type</i>            | <a href="#">plot_type</a>   | <b>Data Plot</b><br>User prompt if plotting grids<br>Batch: fill contours or vectors     | Specifies the type of output plot. By default, this is a simple data plot. To produce a contour or vector plot, use the following plot types:<br><b>Contours:</b> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> </ul> |

|  |                              |   |  |
|--|------------------------------|---|--|
|  |                              |   | <ul style="list-style-type: none"> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <b>Vectors:</b> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>                | <a href="#">plot_scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm=format</b>                      | <a href="#">plot_format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot_param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.  |
| <b>-dr=drawable[,...]</b>              | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.   |
| <b>-la=label[,...]</b>                 | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-in=interval</b>                    | <a href="#">con_interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b>            | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.   |
| <b>-cb=base</b>                        | <a href="#">con_base</a>     | Lowest contour<br>value, maximum<br>wind speed                          | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.   |
| <b>-fl=font_list</b>                   | <a href="#">font_list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.   |
| <b>-ct=color_table</b>                 | <a href="#">color_table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |
| <b>-cot=color[:attr]</b>               | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the text color for labels above and below the plot.  |
| <b>-com=color[:attr]</b>               | <a href="#">color_map</a>    | <b>red</b>  | Specifies the color for background maps.   |
| <b>-coln=color[:attr]</b>              | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.   |
| <b>-cocm=color[:attr]</b>              | <a href="#">color_cmap</a>   | <b>black</b>  | Specifies the color for color fill contour maps.   |
| <b>-coco=color[:attr]</b>              | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for line contours.   |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color_fill</a>   | A predefined<br>color set based<br>on the <b>wxp.clr</b><br>color table | Specifies the color fill contour color set. See the <a href="#">color_fill</a> resource for more details.  |
| <b>-cola=color[:attr]</b>              | <a href="#">color_label</a>  | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the line contour label color.  |
| <b>-cocl=color[:attr]</b>              | <a href="#">color_clabel</a> | <b>black</b>  | Specifies the color fill contour label color.  |
| <b>-de=device[,...]</b>                | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                                    | Specifies the output device.   |
| <b>-ti=title</b>                       | <a href="#">title</a>        | Program name  | Specifies the window title.  |
| <b>-ic</b>                             | <a href="#">icon</a>         | <b>Off</b>  | Specifies whether to open window in iconified mode.  |
| <b>-bg=color</b>                       | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse<br>plots                       | Specifies the background color of the window.  |
| <b>-ge=geometry</b>                    | <a href="#">geometry</a>     | <b>640x480</b>  | Specifies the window and paper geometry.   |
| <b>filename[#seq]</b>                  | <a href="#">filename</a>     | None<br>User Prompt   | The name of the grid data file to be plotted. An optional sequence number can be added to designate the time for non-  |

|  |                                      |
|--|--------------------------------------|
|  | Batch: current= <b>la</b> WXP files. |
|--|--------------------------------------|

## DESCRIPTION

This program contours model gridpoint data. For the most part, this is data stored in GRIB format. This is a compressed binary format ideal for storing model data for operational visualization. This program is set up to adapt to many types of model and file formats. The forecast time, level and variable parameters are set up to mimic GRIB parameters so that any model grid can be queried using **grbcalc**. In addition, there is a way of automatically piecing component grids together prior to display. Then, based on a plot domain, forecast time, level and variable, a contour plot is generated. Overlay plots, contour plots and loops can also be generated if the appropriate options are set.

The program will display data from many model sources. The model type is specified using the **model** resource. The specified string is first looked up in the **model.lup** file which cross-references the model value to a file naming convention. Also, if the model grids are divided up, this lookup file will detail how the grids are to be pieced together. In many cases, global models are split into 4 or even 8 subgrids. Contouring the entire global grid may not be necessary for small domains such as North America. As a result, specialized pieced grids can be developed such as **avn\_na** which will direct **grbcalc** to search and piece together only those grids that affect North America. In other cases, a specific model grid file will contain grids for more than one grid domain. The lookup file can direct **grbcalc** to only use specific grids from a specific domain. This is the case with the ETA model which is broadcast in two grid types: 211 and 207 (for Alaska).

The program starts off by prompting the user for input grid file name. The list of available grid files is based on the file name convention which is dependent on the model type and the file name tag associated with it. The tag can be specifically changed with the **in\_file** resource. The user may specify the input file either via the command line or through the **current** resource.

Next, the program prompts the user for the forecast time. This is done through a forecast time menu which is specified in the **grbcalc.tim** file. Remember, each model type has a specific set of valid forecast times which is defined in the time file. Because of the breadth of models available, not all of the model types will be covered. Even though a menu does not list an appropriate time, it is still accessible by specifically stating the time on either the command line or at the menu prompt.

After the forecast time has been set, the vertical level is prompted for. A level menu is displayed which covers most of the usable levels. This can be tailored with the **grbcalc.lev** file. Again, each model has a set of valid levels and the menu may not give a proper list of levels. If the level is unknown, a value of "miss" can be used and the program will search for the first grid that matches both forecast time and variable type.

The program will now prompt the user for the particular variable to contour. The program will show a menu of commonly used variables but many more are defined internally by the GRIB products and listed in the **grbcalc.var** file. The GRIB variables can be queried with the **griblook** program.

### Sample Internal Variables

| Variable    | Description and units                | Plots |
|-------------|--------------------------------------|-------|
| <b>temp</b> | Temperature (K)                      | value |
| <b>rhum</b> | Relative humidity (%)                | value |
| <b>uwnd</b> | U wind component - EW relative (m/s) | value |
| <b>vwnd</b> | V wind component - NS relative (m/s) | value |
| <b>vvel</b> | Vertical velocity (Pa/s)             | value |
| <b>hght</b> | Geopotential height (m)              | value |
| <b>pres</b> | Pressure (Pa)                        | value |
| <b>prec</b> | Quantitative precipitation (m)       | value |

These parameters can be accessed by name if they have not been redefined in the **grbcalc.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

There are several derived variables from this list which are defined in the **grbcalc.var** file. These include composite plots, overlay plots and computed grids.

### Sample Derived/Composite Variables

| Variable     | Description                                   | Plots  |
|--------------|---|--------|
| <b>wind</b>  | A vector quantity of <uwnd,vwnd>              | vector |
| <b>conv</b>  | Wind convergence (/s) positive is convergence | value  |
| <b>mconv</b> | Moisture convergence (/s)                     | value  |
| <b>rvort</b> | Relative vorticity (/s)                       | value  |
| <b>avort</b> | Absolute vorticity (/s)                       | value  |
| <b>tadv</b>  | Temperature advection (K/s)                   | value  |
| <b>madv</b>  | Moisture advection (/s)                       | value  |
| <b>vadv</b>  | Vorticity advection (/s^2)                    | value  |

The plot domain defaults to the domain of the grid unless changed through the [plot\\_domain](#) resource. The domain can be interactively changed using the mouse once the plot has been generated.

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the grid file is opened and data read in. The piecing of grids will be done at this time. Any additional computations are performed to produce a derived variable.. The background map will plot first unless the output is a color fill contour in which the map plots last. Then, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.
- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed.

When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

## EXAMPLES

```
grbcalc -mo=eta -cu=la -ft=h24 -le=500 -va=rvort -de=d -pl=cf -in=2
```

This contours the 500 mb relative vorticity from the ETA model's 24 hour forecast. The plot is color fill contours with an interval of 2.

```
grbcalc -mo=ngm -cu=la -ft=h24 -le=850 -va=temp -of=wxp -pl=none
```

This will generate grid file based on the **grd** name convention. A specific filename can be used instead of **wxp**.

## FILES

- **grbcalc.tim** - the time menu
- **grbcalc.lev** - the level menu
- **grbcalc.var** - the variable menu and definitions

## SEE ALSO

- [griblook](#) - the GRIB/grid query program
- [contour](#) - the grid contouring program
- [vector](#) - the grid vector plotting program

# GRBSND

## NAME

**grbsnd** - Model GRIB data sounding program

## SYNOPSIS

**grbsnd** [*parameters...*] *filename*

## PARAMETERS

| Command Line                        | Resource                   | Default           | Description  |
|-------------------------------------|----------------------------|-------------------|--|
| <b>-h</b>                           | <a href="#">help</a>       | No                | Lists basic help information.  |
| <b>-df=<i>filename</i></b>          | <a href="#">default</a>    | .wxpdef           | Sets the name of the resource file.  |
| <b>-na=<i>name</i></b>              | <a href="#">name</a>       | grbsnd            | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                          | <a href="#">batch</a>      | No                | Run program in batch mode  |
| <b>-me=<i>level</i></b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• significant and mandatory sounding data - <b>out1a, out1b</b></li> <li>• standard sounding parameters and indices - <b>out1c, out1d</b></li> <li>• parcel information - <b>out2a, out2b</b></li> <li>• wind parameters - <b>out2c</b></li> <li>• wind level data - <b>out3a, out3b</b></li> <li>• wind/storm layer data - <b>out3c</b></li> </ul> |
| <b>-fp=<i>filepath</i></b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.  |
| <b>-dp=<i>datapath</i></b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=<i>conpath</i></b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=<i>rawpath</i></b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-ip=<i>imagepath</i></b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=<i>name_conv</i></b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=<i>in_file</i></b>           | <a href="#">in_file</a>    | grib              | Specifies the input file name tag. This value can also be specified with the <b>model</b> resource by looking up the naming convention in the <b>model.lup</b> file. The default is <b>grib</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-of=<i>out_file[,out...]</i></b> | <a href="#">out_file</a>   | upa_snd           | Specifies what type of raw sounding file to create. By default, no output file is generated when the program. This specifies an output file name tag. The default is <b>upa_snd</b> . A full filename can be specified which can contain name convention tags.   |
| <b>-cu=[<i>hour</i> <b>la</b>]</b>  | <a href="#">current</a>    | None              | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=<i>hour</i></b>              | <a href="#">hour</a>       | None              | This resource specifies the exact hour that a data file is valid   |

|   |                                |  |   |
|---|--------------------------------|--|---|
|   |                                |  | for. This locks in the start hour for a multi-file sequence.  |
| <b>-mo=</b> <i>model</i>                                    | <a href="#">model</a>          | <b>ngm</b> or first model listed in <b>model.lup</b> file. | Specifies the model to use as input data. These are defined in the <b>model.lup</b> file. This file can define how model grids are pieced together. Some samples are: <ul style="list-style-type: none"> <li>• <b>ngm</b> -- Nested Grid Model</li> <li>• <b>eta</b> -- ETA model</li> <li>• <b>avn</b> -- Aviation model</li> </ul>  |
| <b>-ft=</b> <i>time</i>                                     | <a href="#">time</a>           | User prompt<br>Batch: initial time                         | This specifies the forecast time to be used in the plots. To create loops, a list of forecast time are specified separated by commas.   |
| <b>-id=</b> <i>identifier</i><br><b>-id=</b> <i>lat,lon</i> | <a href="#">Identifier</a>     | None   | Specifies the station or location to use in the program. If a station is specified, its location is queried from <b>sao.cty</b> database. A specific latitude and longitude can be specified as <i>lat,lon</i> .  |
| <b>-pd=</b> <i>domain</i>                                   | <a href="#">plot_domain</a>    | None.<br>Full plot   | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-pl=</b> <i>plot_type</i>                                | <a href="#">plot_type</a>      | User prompt<br>Batch: <b>skewt</b>                         | Specifies the type of output thermodynamic plot. By default, this is a Skew T diagram. The available plot types are: <ul style="list-style-type: none"> <li>• <b>skewt</b> -- Skew T log P diagram</li> <li>• <b>ema</b> -- Emagram</li> <li>• <b>stuve</b> -- Stuve diagram</li> <li>• <b>hodo</b> -- Hodogram</li> <li>• <b>none</b> -- for no output plot</li> </ul>   |
| <b>-sc=</b> <i>scale_factor</i>                             | <a href="#">plot_scale</a>     | <b>1.0</b>   | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=</b> <i>format</i>                                   | <a href="#">plot_format</a>    | None   | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.  |
| <b>-pp=</b> <i>param</i><br><i>[,param...]</i>              | <a href="#">plot_param</a>     | None   | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=</b> <i>drawable</i> [,...]                          | <a href="#">draw</a>           | <b>all</b>   | Specifies which part of the plot is to be drawn.  |
| <b>-la=</b> <i>label</i> [,...]                             | <a href="#">label</a>          | Program defaults   | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.   |
| <b>-fl=</b> <i>font_list</i>                                | <a href="#">font_list</a>      | <b>modern.fnt</b>  | Specifies a list of fonts to use in the plots.  |
| <b>-ct=</b> <i>color_table</i>                              | <a href="#">color_table</a>    | <b>wxp.clr</b>   | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=</b> <i>color</i> [: <i>attr</i> ]                  | <a href="#">color_text</a>     | <b>white</b><br><b>black</b> for inverse plots.            | Specifies the text color for labels above and below the plot.   |
| <b>-coth=</b> <i>color</i> [: <i>attr</i> ]                 | <a href="#">color_therm</a>    | <b>blue, blue,</b><br><b>green, cyan,</b><br><b>yellow</b> | This resource is used to specify the color and attributes for the thermodynamic plot. There are 5 possible values: <ul style="list-style-type: none"> <li>• pressure lines (default <b>blue</b>)</li> <li>• temperature lines (default <b>blue</b>)</li> <li>• potential temperature lines -- dry adiabats (default <b>green</b>)</li> <li>• equivalent potential temperature lines -- moist adiabats (default <b>cyan</b>)</li> <li>• mixing ratio lines (default <b>yellow</b> or <b>brown</b> on inverse color plots)</li> </ul> |
| <b>-cott=</b> <i>color</i> [: <i>attr</i> ]                 | <a href="#">color_thermtxt</a> | <b>blue, lblue, off,</b><br><b>off, yellow</b>             | This resource is used to specify the color and attributes for the labels on the thermodynamic plot. There are 5 possible values: <ul style="list-style-type: none"> <li>• pressure lines (default <b>blue</b>)</li> </ul>   |

|                                    |                              |   |   |
|------------------------------------|------------------------------|---|---|
|                                    |                              |   | <ul style="list-style-type: none"> <li>temperature lines (default <b>blue</b>)</li> <li>potential temperature lines -- dry adiabats (default <b>off</b> - does not plot normally)</li> <li>equivalent potential temperature lines -- moist adiabats (default <b>off</b> - does not plot normally)</li> <li>mixing ratio lines (default <b>yellow</b> or <b>brown</b> on inverse color plots)</li> </ul>   |
| <b>-cosd=</b> <i>color[:attr]</i>  | <a href="#">color_sound</a>  | <b>white, white, yellow</b>                     | This resource is used to specify the color and attributes for the sounding plot. There are 3 possible values: <ul style="list-style-type: none"> <li>temperature sounding (default <b>white</b> or <b>black</b> for inverse plots)</li> <li>dewpoint sounding (default <b>white</b> or <b>black</b> for inverse plots)</li> <li>parcel sounding (default <b>yellow</b> or <b>brown</b> on inverse color plots)</li> </ul>   |
| <b>-cowsd=</b> <i>color[:attr]</i> | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse plots. | Specifies the color for wind barbs on thermodynamic plots.  |
| <b>-coho=</b> <i>color[:attr]</i>  | <a href="#">color_hodo</a>   | <b>blue, blue, white, cyan, red</b>             | This resource is used to specify the color and attributes for the hodograph plot. There are 5 possible values: <ul style="list-style-type: none"> <li>direction lines (default <b>blue</b>)</li> <li>speed circles (default <b>blue</b>)</li> <li>sounding line up to 700 mb (default <b>white</b> or <b>black</b> for inverse plots)</li> <li>sounding line above 700 mb (default <b>cyan</b>)</li> <li>marker "+" denoting each level (default <b>red</b>)</li> </ul> |
| <b>-coht=</b> <i>color[:attr]</i>  | <a href="#">color_hodtxt</a> | <b>lblue, lblue, lred</b>                       | This resource is used to specify the color and attributes for the labels on the hodograph plot. There are 3 possible values: <ul style="list-style-type: none"> <li>direction labels (default <b>lblue</b>)</li> <li>speed labels (default <b>lblue</b>)</li> <li>sounding labels denoting pressure level (default <b>lred</b>)</li> </ul>  |
| <b>-de=</b> <i>device[,...]</i>    | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>            | Specifies the output device.  |
| <b>-ti=</b> <i>title</i>           | <a href="#">title</a>        | Program name                                    | Specifies the window title.   |
| <b>-ic</b>                         | <a href="#">icon</a>         | <b>Off</b>                                      | Specifies whether to open window in iconified mode.   |
| <b>-bg=</b> <i>color</i>           | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots  | Specifies the background color of the window.   |
| <b>-ge=</b> <i>geometry</i>        | <a href="#">geometry</a>     | <b>640x480</b>                                  | Specifies the window and paper geometry.  |
| <i>filename[#seq]</i>              | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the upper air converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files.  |

## DESCRIPTION

This program plots upper air soundings based on model data. The output is very similar to the **uacalplt** program and plots a thermodynamic diagram or hodograph. The input to the program is model data stored in GRIB format. The program will display data from many model sources. The model type is specified using the [model](#) resource. The specified string is first looked up in the **model.lup** file which cross-references the file to a file naming convention.

The programs starts off by prompting the user for input grid file name. The list of available grid files is based on the file name convention which is dependent on the model type and the file name tag associated with it. The tag can be specifically changed with the [in file](#) resource. The user may specify the input file either via the command line of through the [current](#) resource.

Next, the program prompts the user for the forecast time. This is done through a forecast time menu which is specified in the **grbsnd.tim** file. Remember, each model type has a specific set of valid forecast times which is defined in the time file. Because of the breadth of models available, not all of the model types will be covered. Even though a menu does not list an appropriate time, it is still accessible by specifically stating the time on either the command line or at the menu prompt.

Next, the user enters a station identifier. If a station identifier is specified, its location is queried from **sao.cty** database. A specific latitude and longitude can be specified as *lat,lon*.

Once the station is selected, the program retrieves the grids from the model file. These include temperature, relative humidity, U and V wind components and height for all levels from 1000 mb to 100 mb. For most models, this data is available every 50 mb. The data is then interpolated to the station location and a sounding is constructed. Once the sounding is available, the program prints out sounding information, and standard sounding parameters and indices.

## Text Output

The first section is the station location information which is printed at message level **out1a**:

```
Date: 24 hour ETA valid 0Z WED 1 OCT 97
Station: KIND
Latitude: 39.73
Longitude: -86.27
```

The next section is the raw sounding data. This is a merged set of mandatory and significant level data. Additional computations are performed. Here is a list of the columns printed (all printed at mess level **out1a** [see red text] unless specified)

- LEV -- level number. Special levels are denoted with SFC (surface), TRP (tropopause), WND (max wind).
- PRES -- pressure of level in mb
- HGHT -- geopotential height in m (intermediate levels interpolated using hypsometric approximation)
- TEMP -- temperature in C
- DEWP -- dewpoint in C
- RH -- relative humidity in %
- DD -- dewpoint depression in C (mess level **out1b**)
- WETB -- wetbulb temperature in C (mess level **out1b**)
- DIR -- wind direction in deg (intermediate levels interpolated using significant wind level data)
- SPD -- wind speed in knt (intermediate levels interpolated using significant wind level data)
- THETA -- potential temperature in K (mess level **out1b**)
- THE-V -- virtual potential temperature in K (mess level **out1b**)
- THE-W -- wetbulb potential temperature in K (mess level **out1b**)
- THE-E -- equivalent potential temperature in K (mess level **out1b**)

```
-----
LEV PRES  HGHT  TEMP  DEWP  RH  DD  WETB  DIR  SPD  THETA  THE-V  THE-W  THE-E  W
      mb      m      C      C      %  C      C  deg  knt  K      K      K      K      g/kg
-----
0 1000  109
SFC 986  228  15.6  10.1  70  5.5  12.4  316  9 289.9  291.3  286.1  312.2  7.88
2 950  545  12.9  8.8  76  4.1  10.6  315  19 290.3  291.6  285.9  311.5  7.49
3 900  997  8.7  7.1  89  1.7  7.8  319  21 290.5  291.7  285.5  310.5  7.02
4 850  1467  5.1  3.2  88  1.9  4.1  328  21 291.5  292.5  284.5  307.8  5.68
5 800  1960  3.4  -2.8  64  6.2  0.6  338  20 294.8  295.5  283.9  306.3  3.90
...

```

The next section displays standard sounding parameters and indices. Most of the information is printed at message level **out1d** except for thickness, lifted, Showalter, total totals and K indices which is at message level **out1c** (red text).

#### Sounding variables and indices

```

1000-500 mb thick:      5587.03 m
Freezing level:        643.64 mb = 3714.58 m = 12186.81 ft
Wetbulb zero:          789.00 mb = 2072.41 m = 6799.17 ft
Precipitable water:    0.67 inches
Sfc-500 mean rel hum:  41.53 %
Est. max temperature:  18.05 C = 64.49 F
Sfc-Lift cond lev (LCL): 908.75 mb = 915.84 m = 3004.69 ft T: -263.68 C
700-500 lapse rate:   4.61 C/km
ThetaE index:          5.84 C Layer 950.0- 750.0 mb
Conv cond level (CCL): 881.54 mb = 1167.24 m = 3829.49 ft
  Mean mixing ratio:    7.34 g/kg
  Conv temperature:     16.54 C = 61.78 F
Cap Strength:          9.95 C
Lifted Index:          10.17 C Risk: None
Lifted Index @300 mb:  10.90 C
Lifted Index @700 mb:  4.37 C
Showalter Index:       12.28 C Risk: None
Total Totals Index:    31.16 C Risk: None
  Vertical Totals Index: 16.51 C
  Cross Totals Index:   14.65 C
K Index:               4.27 Risk: None
Sweat Index:           138.97 Risk: None
Energy Index:          2.72 Risk: None

```

The next section involves a parcel trajectory. The parcel is taken from some initial condition and raised vertically to the top of the sounding. The default parcel uses a 100 mb mean layer above the surface. The parcel type can be changed with the **par** plot parameter:

**par**=pres:temp:dew

Temperature and dewpoint are optional. If not specified, the sounding temperature and dewpoint are used.

| Type            | Pres         | Temp         | Dew        |
|-----------------|--------------|--------------|------------|
| pressure level  | <b>pres</b>  | <i>temp</i>  | <i>dew</i> |
| surface level   | <b>sfc</b>   | <i>temp</i>  | <i>dew</i> |
| max temperature | <b>maxt</b>  | --           | --         |
| pressure layer  | <b>layer</b> | <i>thick</i> | --         |

The data are printed at message level **out2b** except for CAPE, convective inhibition and cap strength which is printed at level **out2a** (red text).

#### Parcel Indices

Parcel: using 100 mb layer

```

CAPE (B+):              5.86 J/kg
Max Up Vert Vel:        3.42 m/s
Conv Inhibition (B-):   3.15 J/kg
Cap Strength:           10.12 C
Lift Cond Lev (LCL):    888.66 mb = 1101.03 m = 3612.25 ft
Lev Free Conv (LFC):    863.66 mb = 1335.91 m = 4382.86 ft
Equ Level (EL):         823.66 mb = 1723.24 m = 5653.59 ft
B at Equ Level:         5.60 J/kg
Max Parcel Lev (MPL):   793.66 mb = 2024.75 m = 6642.80 ft

```

The next section displays significant wind level data. Storm relative winds are computed based on a mean storm motion. This is computed using the mean wind of the lower 6km of the atmosphere. The storm direction is deflected

30 deg clockwise and the speed is 75% of the 6km mean speed. The output is at message level **out3a** (red text) except for the storm relative winds which is at **out3b**.

```

Wind level data
Storm motion: 349 at 27 knt
-----
  LEV  P    H          TRUE/GROUND REL      STORM REL
      mb   m          DIR SPD  U    V      DIR SPD  U    V
                        knt   m/s      knt   m/s
-----
  0 1000  109
  1  986  228  316   9   6.1  -6.4 | 183  20   1.1  20.4
  2  950  545  315  19  13.2 -13.3 | 211  16   8.2  13.5
  3  900  997  319  21  14.0 -15.9 | 220  14   9.1  10.9
  4  850 1467  328  21  11.2 -17.6 | 214  11   6.2   9.1
  ...

```

The next section is wind parameters and indices. This is printed at message level **out2d** except for mean winds, storm direction, helicity and energy-helicity index which are at level **out2c** (red text)

```

Wind Parameters
Mean winds (0-6000m):          319.5 at 36.3 knts
Storm direction:              349.5 at 27.2 knts
Shear (0-3000m)               pos: 2.1 neg: 2.5 tot: 4.6 10^-3/s
Storm rel Dir Shear (0-3000m): 87.9 deg
Storm rel helicity (0-3000m) pos: 78.0 neg: -5.8 tot: 72.2 m^2/s^2
                              ave: 24.1 10^-3 m/s^2 rel: 0.63
Storm rel vorticity (0-3000m) horiz: 5.6 stream: 3.5 10^-3/s
Energy-Hel index:             0.00
Bulk Rich Number:             0.10
Bulk Shear:                   56.19 m/s

```

The next section shows storm layer parameters. All of this information is printed at message level **out3c**.

```

Storm Parameters
-----
Depth  Mean   Inflow|   Shear          Helicity          Vorticity
AGL   Dir Spd Dir Spd| Pos  Neg  Tot| Pos  Neg  Tot  Ave  Rel| Hor Strm
m      knt  knt  | 10^-3/s      m^2/s^2      m/s^2      | 10^-3/s
-----
 500  316  16 201 17| 11.2 0.0 11.2| 42.7 0.0 42.7 85.4 0.88| 11.4 10.0
1000  318  18 209 15| 6.3 0.0 6.3 | 45.4 0.0 45.4 45.4 0.76| 7.1 5.4
1500  321  19 210 14| 4.3 0.0 4.3 | 45.4 -5.8 39.6 26.4 0.48| 6.0 2.9
2000  325  19 210 12| 3.2 0.8 4.0 | 51.9 -5.8 46.1 23.0 0.55| 5.3 2.9
2500  326  21 215 12| 2.6 1.9 4.4 | 65.0 -5.8 59.2 23.7 0.61| 5.6 3.4
3000  325  23 224 11| 2.1 2.5 4.6 | 78.0 -5.8 72.2 24.1 0.63| 5.6 3.5
4000  324  27 246 12| 1.6 3.3 4.9 |107.6 -5.8 101.8 25.4 0.59| 5.7 3.4
5000  322  32 263 15| 1.3 3.7 5.0 |121.3 -6.9 114.4 22.9 0.51| 5.6 2.9
6000  319  36 273 19| 1.1 3.5 4.6 |121.3 -42.1 79.2 13.2 0.40| 5.3 2.1

```

## Sounding Plots

After the text is displayed, the user is prompted for the output plot type. The selections are:

- **skewt** -- Skew T log P diagram
- **ema** -- Emagram
- **stuve** -- Stuve diagram
- **hodo** -- Hodogram

A thermodynamic diagram is plot showing the temperature and dewpoint plotted versus pressure (or height). A hodograph is a plot of wind speed versus direction on a polar grid.

**Skew-T log-P:** The skew-T log-P diagram is an energy conserving diagram in which the temperature and potential temperature lines are nearly perpendicular. The pressure lines are plotted horizontally in blue and are also on an inverse log scale. The concept of Skew T means that the temperature is not plotted vertically but angles off to the right at a 45 degree angle. The temperature lines of the Skew T are in blue. The dry adiabats slope off to the upper left and are plotted in green. The light blue dashed curved lines are saturation adiabats. The yellow dashed lines are lines of constant mixing ratio.

The sounding is displayed in white, with the white line on the left of the plot representing the dewpoint line and the white data line on the right representing the temperature line. When printed, these are bold lines for easier viewing. The wind barbs on the right-hand side of the diagram represent the winds at various elevations, with the top of the diagram representing north. The winds at mandatory levels are plotted at their pressure levels, whereas the winds at height levels are plotted at their standard atmosphere pressure level.

**Emagram:** An emagram is very close to a skewT except the temperature lines are vertical, not skewed to the right. It is also an energy conserving diagram.

**Stuve:** In the Stuve diagram format, the diagram is much the same as the Skew T plot except that the temperature lines are vertical. Pressure lines are spaced so that potential temperature lines are straight. This is not an energy conserving diagram.

**Wind plot:** The winds at selected levels are plotted on the right side of the diagram. Standard directions are used with up being north and right being to the east. These are plotted in white but can be changed with the **color\_wind** resource.

**Data plot:** The geopotential heights are plotted at a 50 mb interval on the left side of the plot. The color of the data can be changed with the **color\_wind** resource. On the right side of the plot, specific parameters and indices are plotted. These values are:

Station information:

|            |                                |
|------------|--------------------------------|
| <b>WMO</b> | WMO number                     |
| <b>LAT</b> | Latitude of station            |
| <b>LON</b> | Longitude of station           |
| <b>EL</b>  | Elevation of station in meters |

Key information:

|             |   |
|-------------|---|
| <b>TP</b>   | Tropopause level (mb)                               |
| <b>MW</b>   | Max wind level (mb)                                 |
| <b>FRZ</b>  | Lowest freezing level (mb) or BG for below ground   |
| <b>WB0</b>  | Wet bulb zero (mb) or BG                            |
| <b>PW</b>   | Precipitable water (in)                             |
| <b>RH</b>   | Mean RH surface to 500 mb (%)                       |
| <b>MAXT</b> | Estimated max temperature (C) using a 150mb layer   |
| <b>TH</b>   | 1000-500mb thickness (m)                            |
| <b>L57</b>  | 700-500mb lapse rate (C/km)                         |
| <b>LCL</b>  | Lift condensation level (mb) from surface data      |
| <b>LI</b>   | Lifted index (C) using 100 mean layer above surface |
| <b>SI</b>   | Showalter index (C)                                 |
| <b>TT</b>   | Total totals index                                  |
| <b>KI</b>   | K index   |
| <b>SW</b>   | Sweat index   |
| <b>EI</b>   | Energy index  |

**-PARCEL-**

This is a parcel trajectory (the yellow line on the sounding) based on 100 mb mean layer.

|             |                                       |
|-------------|---------------------------------------|
| <b>CAPE</b> | Convective available potential energy |
| <b>CINH</b> | Convective inhibition (open ended)    |
| <b>LCL</b>  | Lift condensation level (mb)          |
| <b>CAP</b>  | Cap strength (C)                      |
| <b>LFC</b>  | Level free convection (mb)            |
| <b>EL</b>   | Equilibrium level (mb)                |
| <b>MPL</b>  | Maximum parcel level (mb)             |

**-WIND-**

Wind parameters

|             |   |
|-------------|---|
| <b>STM</b>  | Estimated storm motion (knts) from 0-6000m AG mean layer, spd 75% of mean, dir 30 deg veer from mean. |
| <b>HEL</b>  | Storm relative helicity 0-3000m AG (total value)  |
| <b>SHR+</b> | Positive shear magnitude 0-3000m AG (sum of veering shear values)                                     |
| <b>SRDS</b> | Storm relative directional shear 0-3000m AG (directional difference of storm relative winds)          |
| <b>EHI</b>  | Energy helicity index (prop to positive helicity * CAPE)  |
| <b>BRN</b>  | Bulk Richardson number 500-6000m AG (prop to CAPE/bulk shear)   |
| <b>BSHR</b> | Bulk shear value (magnitude of shear over layer)  |

The amount of data plotted can be regulated by using the **data** plot parameters:

- **nodata** -- no data plotted
- **data** -- all data plotted
- **data1** -- minimal data plotted
- **data2** -- important data plotted

**Parcel Trajectory:** A parcel trajectory is plotted with each sounding as a thick yellow line. If no parcel trajectory is needed, specifying the **noparcel** plot parameter will eliminate it from the plot. The type of parcel can be changed by specifying the **par** plot parameter.

**Wind hodograph:** The wind hodograph is a polar plot of wind speed and direction. The radius directly reflects wind speed. The hodograph is plotted for all mandatory and wind at height levels with the plot annotated with plus signs at each level and labeled with the mandatory level pressures above and to the right of the plus sign. The lowest level data (below 700 mb) is plotted in white.

The plot type can be changed. If the thermodynamic diagram is visible, the hodograph can be plotted by hitting <Meta>n. By hitting <Meta>n again, the thermodynamic plot will reappear.

**Sounding Modification**

Once the data are plotted, the sounding can be modified. This can be done by moving the cursor to the line which is to be modified, holding down the <Shift-right button> and moving the cursor to where the new value will be. If the cursor is right of the temperature line, the temperature sounding is modified. If the cursor is left of the dewpoint line, the dewpoint is modified. This can only be done on the existing levels. This can also be done on the hodograph.

In many cases, the sounding has too much detail to effectively select a level or data point. It is recommended that the user zoom the plot by using the <Shift-left button> and dragging a box around the area to be zoomed. Remember to start at a corner.

Once the sounding is modified, the plotted data will update to reflect the changed sounding. The parcel will not automatically update. This can be done with the <Meta>r refresh command. The modified sounding can be saved to a raw file using the <Meta>s save command. This can be read in by **uacalplt** to be replotted.

**Output Files**

By default, this program does not produce output files. A raw file can be generated using the <Meta>s command in the plot window. The **out\_file** resource specifies a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

## EXAMPLES

```
grbsnd -cu=la -mo=eta -ft=h24 -id=KIND -pl=skewt -de=d
```

This plots a 24 hour ETA forecast skewT diagram for KIND.

```
grbsnd -cu=la -mo=ngm -ft=h30 -id=30,-100 -pl=skewt -de=d
```

This plots a 30 hour NGM forecast skewT diagram for the location 30N, 100W.

## FILES

- **sao.cty** - the surface station location database file

## SEE ALSO

- [grbcalc](#) - the model grid contouring program
- [uacalplt](#) - the upper air sounding plotting program

# GRDMATH

## NAME

**grdmath** - Mathematical manipulation program for grids

## SYNOPSIS

**grdmath** [*parameters...*] *operations*

## PARAMETERS

| Command Line                    | Resource                   | Default                        | Description   |
|---------------------------------|----------------------------|--------------------------------|---|
| <b>-h</b>                       | <a href="#">help</a>       | No                             | Lists basic help information.   |
| <b>-df=filename</b>             | <a href="#">default</a>    | .wxpdef                        | Sets the name of the resource file.   |
| <b>-na=name</b>                 | <a href="#">name</a>       | grdmath                        | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                      | <a href="#">batch</a>      | No                             | Run program in batch mode   |
| <b>-me=level</b>                | <a href="#">message</a>    | out2                           | Specifies level of messages to be displayed   |
| <b>-fp=filepath</b>             | <a href="#">file_path</a>  | current directory              | Specifies location of database files.   |
| <b>-gp=gridpath</b>             | <a href="#">grid_path</a>  | current directory              | Specifies the location/path of input and output grid files.   |
| <b>-gt=grid_type</b>            | <a href="#">grid_type</a>  | first grid                     | Specifies the grid type number from the GRIB file. If none is specified, only the first grid is used.   |
| <b>-of=out_file</b>             | <a href="#">out_file</a>   | wxp                            | Specifies what type of output grid file the program will produce. The input file type is queried when the file is opened but the user may change the output file type. The options are: <ul style="list-style-type: none"> <li>• <b>wxp</b> - ASCII WXP grid file format</li> <li>• <b>wxpb</b> - Binary WXP grid file format</li> <li>• <b>cdf</b> - netCDF grid file format</li> </ul>  |
| <b>-pf=file_param</b>           | <a href="#">file_param</a> | prompt user, use in batch mode | Specifies how to deal with writing to files that already exist. Possible values are: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program sometimes terminates when this occurs. This is especially true for decoders which perform no other operation than to create this file.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> </ul>  |
| <b>-pa=param</b><br>[,param...] | <a href="#">parameter</a>  | None                           | Extra parameters (reserved for future use)  |
| <i>command</i>                  | <a href="#">command</a>    | None                           | The command resource specifies the files and values to use in the mathematical operations along with the function to perform. The commands are of the following formats: <ul style="list-style-type: none"> <li>• <b>func</b> <i>F1 F2 ... = FO [Type] [Date]</i></li> <li>• <i>F1 oper</i> <b>F2 = FO [Type] [Date]</b></li> </ul> The file specifications ( <i>F1</i> , <i>F2</i> ) can either be a filename or a number. The filename can be specified with an extension to define a specific grid within a multiple grid file: <i>filename[+time:level:var[:units]]</i><br>The time, level, var, unit definitions are listed in the description section (also see <a href="#">filename</a> resource).<br>The number must be preceded number sign "#" as in "#5.6".<br><i>FO</i> is the specific output filename. All filenames are prepended with the <a href="#">grid_path</a> resource value unless a full path is specified. |

|  |  |  |   |
|--|--|--|---|
|  |  |  | <i>Type</i> and <i>Date</i> are optional strings used to modify the information header of the grid file. This is useful in making sure the resulting grid has the proper information since the math functions often misrepresent the final data type. |
|--|--|--|---|

## DESCRIPTION

The grdmath program performs mathematical operation on gridpoint data. The functions and the filenames are interpreted from the command line as part of the **command** resource. There are two syntax's used.

### Algebraic Format

The first form uses an algebraic approach:

$$F1 \text{ oper} F2 = FO [Type] [Date]$$

where the operator (**oper**) is a simple algebraic operator:

- **add**
- **sub**
- **mul**
- **div**
- **mod** - modulus operator (remainder of A/B)

The values *F1* and *F2* represent either filenames or numeric values. The numeric values have the following syntax:

*#number*  
or  
*#cor*

The number listed can be any integer or floating point number. This option also allows the specification of the Coriolis force (**#cor**). This value is calculated for each grid point and added, subtracted, etc based on the math function.

The filenames have the following syntax:

*filename*[+*time*:*level*:*var*[:*units*]]  
or  
*filename*[#*num*[:*units*]]

where *filename* is the name of the grid file to use. If a relative path is used, the value of the **grid\_path** is prepended to the filename. The filename can be "**std**" where standard input is used.

The extra information following the "+" or "#" defines which grid to use within a multigrid file. The value of *num* represents the sequence number of the grid in the file. Grids are numbered starting at 0 (zero). This number can be determined through a **griblook** listing.

The values of *time*, *level*, and *var* make it possible to specify the grid based on typical grid information. Possible values for each are:

- **time:** init,06,12,18,24,30,36,42,48,60,72,d3,d4,d5,d6,d7,d8,d9,d10
- **level:** sfc, sl (sea level), trop, wind, 1000, 850, 700, 500, 400, 300, 200, 100
- **var:** temp, rhum, pres, vvel, uwnd, vwnd, prec, lift

The *unit* specification allows the user to change the units of the grid before performing the math.

The *FO* specifies the output filename which stores the result of the calculation. If a relative path is specified, the value of the **grid\_path** is prepended to the filename. The filename can be "**std**" where standard output is used.

The *Type* specification is optional but can be used to set the new data type in the grid file information header. When the grid is contoured, the data type is put in the upper left hand corner of the plot. The **grdmath** program will attempt to create a new data type label but many times this is incorrect. This allows the user to correct the output data type.

The *Date* specification is optional but just like the *Type* specification if the result of the calculations is incorrect.

## Function Format

**func** *F1 F2 ... = FO [Type] [Date]*

Much of the specification for this format is the same as in the algebraic format. The difference is that there are more functions and some functions take more than 2 filenames/values. Here is a list of the functions and their required fields:

**QUERY** - queries information from the grid, no grid output

- **info file**  
This prints out a 4 line report on the contents of the grid including data type and grid domain.
- **domain file**  
Prints only the grid domain in a format compatible with the **plot\_domain** resource.
- **val file id ID**  
Prints the value interpolated from the grid based on the location of a specified station *ID*
- **val file ll lat,lon**  
Prints the value interpolated from the grid based on a specified latitude and longitude
- **val file x,y**  
Prints the value interpolated from the grid based on a specified gridpoint location (0,0 is up, right)
- **loc file id ID**  
Prints the gridpoint location of the specified station *ID*
- **loc file ll lat,lon**  
Prints the gridpoint location of the specified latitude and longitude

**STATISTICS** - prints values, no grid output

- **max file**  
Prints the maximum value of the grid
- **min file**  
Prints the minimum value of the grid
- **maxmin file**  
Prints both the max and the min of the grid
- **avg file**  
Prints the average value of the grid
- **avgnum file**  
Prints the average value of the grid along with the number of valid gridpoints

**MANIPULATION** - produces hybrid grids

- **copy file1 = ofile**  
Copies the initial grid field in *file1* to the output file. This is useful when converting units or extracting a single grid from a multigrid file.
- **piece nxxny file1 file2 ... = ofile**  
This pieces together multiple grids into one composite grid. The number of grids to piece are specified by the *nx* and *ny* parameters. The component grids start with *file1* in the upper right corner. The next grid is placed just to the right of the first grid in the first row (if *nx* > 1). The piecing continues until the first row is complete where the next row is then filled.
- **remap file plot\_domain = ofile**

This will remap a grid using simple interpolation to a new grid as specified in the *plot\_domain* specification. *ofile* will contain the remapped grid.

**MATHEMATICS** - performs math functions and returns a grid

- **max** *file1 file2 ... = ofile*  
Computes the maximum value at each gridpoint based on the values in the files *file1*, *file2*, etc. More than two files can be listed
- **min** *file1 file2 ... = ofile*  
Computes the minimum value at each gridpoint based on the values in the files *file1*, *file2*, etc. More than two files can be listed
- **sum** *file1 file2 ... = ofile*  
Computes the sum of each gridpoint based on the values in the files *file1*, *file2*, etc. More than two files can be listed.
- **add** *file1 file2 ... = ofile*  
Same as **sum**
- **avg** *file1 file2 ... = ofile*  
Computes the average at each gridpoint based on the values in the files *file1*, *file2*, etc. More than two files can be listed
- **interp** *file1 file2|value file3 = ofile*  
This interpolates between the grid specified in *file1* and the grid in *file3* based on the value specified either in *file2* or the *value*. This does a simple linear interpolation. A value of **.5** would essentially give an average. A value of **.8** will return a value that is **.8** of the way to the second grid.
- **sub** *file1 file2 = ofile*  
This subtracts *file2* from *file1*.
- **thick** *file1 file2 = ofile*  
Same as **sub**
- **mul** *file1 file2 = ofile*  
Multiplies the grids in *file1* and *file2*.
- **div** *file1 file2 = ofile*  
Divides the values in *file1* by the values in *file2*.
- **mod** *file1 file2 = ofile*  
Does the modulus of *file1* divided by *file2*. This produces the remainder of the division computation.
- **sqrt** *file1 = ofile*  
Does the square root of all values in *file1*.
- **abs** *file1 = ofile*  
Returns the absolute value of the values in *file1*.
- **log** *file1 = ofile*  
Takes the logarithm of each value in *file1*
- **log10** *file1 = ofile*  
Takes the logarithm base 10 of each value in *file1*
- **exp** *file1 = ofile*  
Exponentiates ( $e^x$ ) the values in *file1*. This is the inverse of the log function
- **pow** *file1 file2 = ofile*  
This performs for power function of the values in *file1* raised to the power of the values in *file2*.
- **sin** *file1 = ofile*  
Performs the trigonometric function sine of the values in *file1*. Input values are in degrees
- **cos** *file1 = ofile*  
Performs the trigonometric function cosine of the values in *file1*. Input values are in degrees
- **tan** *file1 = ofile*  
Performs the trigonometric function tangent of the values in *file1*. Input values are in degrees
- **asin** *file1 = ofile*  
Performs the trigonometric function arcsine of the values in *file1*. Output values are in degrees
- **acos** *file1 = ofile*  
Performs the trigonometric function arccosine of the values in *file1*. Output values are in degrees

- **atan** *file1* = *ofile*  
Performs the trigonometric function arctangent of the values in *file1*. Output values are in degrees
- **atan2** *file1 file2* = *ofile*  
Performs the trigonometric function arctangent of the values in *file1* divided by *file2*. The sign of 2 values determines the quadrant. This is the equivalent of the C atan2 function. Output values are in degrees
- **dewp** *file1 file2* = *ofile*  
Computes the dewpoint based on temperature in *file1* and the relative humidity in *file2*
- **rhum** *file1 file2* = *ofile*  
Computes the relative humidity based on temperature in *file1* and the dewpoint in *file2*
- **wetblb** *file1 file2* = *ofile*  
Computes the wetbulb temperature based on temperature in *file1* and the relative humidity or dewpoint in *file2*.
- **wchill** *file1 file2 [file3]* = *ofile*  
Computes the wind chill temperature based on the temperature in *file1* and wind speed in *file2*, or the U wind component in *file2* and the V wind component in *file3*.
- **heat** *file1 file2* = *ofile*  
Computes the heat index temperature based on temperature in *file1* and the relative humidity or dewpoint in *file2*.
- **theta** *file1 [file2]* = *ofile*  
Computes the potential temperature based on temperature in *file1* and the pressure in *file2*. The pressure is optional if *file1* is on a pressure surface (i.e. 500mb).
- **thetac** *file1 file2 [file3]* = *ofile*  
Computes the equivalent potential temperature based on temperature in *file1*, the relative humidity or dewpoint in *file2* and the pressure in *file3*. The pressure is optional if *file1* is on a pressure surface (i.e. 500mb).
- **thetav** *file1 file2 [file3]* = *ofile*  
Computes the virtual potential temperature based on temperature in *file1*, the relative humidity or dewpoint in *file2* and the pressure in *file3*. The pressure is optional if *file1* is on a pressure surface (i.e. 500mb).
- **vtemp** *file1 file2 [file3]* = *ofile*  
Computes the virtual temperature based on temperature in *file1*, the relative humidity or dewpoint in *file2* and the pressure in *file3*. The pressure is optional if *file1* is on a pressure surface (i.e. 500mb).
- **vapor** *file1 [file2]* = *ofile*  
Computes the vapor pressure based on dewpoint in *file1*, or the temperature in *file1* and the relative humidity in *file2*.
- **shum** *file1 [file2] [file3]* = *ofile*  
Computes the specific humidity based on dewpoint in *file1* and pressure in *file2*, or temperature in *file1*, relative humidity in *file2* and pressure in *file3*. The pressure is optional if *file1* is on a pressure surface (i.e. 500mb).
- **mixrat** *file1 [file2] [file3]* = *ofile*  
Computes the mixing ratio based on dewpoint in *file1* and pressure in *file2*, or temperature in *file1*, relative humidity in *file2* and pressure in *file3*. The pressure is optional if *file1* is on a pressure surface (i.e. 500mb).
- **spd** *file1 file2* = *ofile*  
Computes the wind speed based on U wind component in *file1* and the V wind component in *file2*.
- **mag** *file1 file2* = *ofile*  
Computes the vector magnitude based on X component in *file1* and the Y component in *file2*.
- **dir** *file1 file2* = *ofile*  
Computes the vector/wind direction based on X/U wind component in *file1* and the Y V wind component in *file2*.
- **ugeos** *file1* = *ofile*  
Computes the U geostrophic wind component based on geopotential height in *file1*.
- **vgeos** *file1* = *ofile*  
Computes the V geostrophic wind component based on geopotential height in *file1*.
- **uq** *file1 file2* = *ofile*  
Computes the U Q-vector component based on geopotential height in *file1* and temperature in *file2*.
- **vq** *file1 file2* = *ofile*

- Computes the V Q-vector component based on geopotential height in *file1* and temperature in *file2*.
- **dx** *file1* = *ofile*  
Finite differences (**dfile1/dx**) the grid *file1* in the X direction.
- **dy** *file1* = *ofile*  
Finite differences (**dfile1/dy**) the grid *file1* in the Y direction.
- **grad** *file* = *xfile* *yfile*  
Gradient of the grid. Outputs 2 grids, one for the X component and the second for the Y component.
- **vdiff** *xfile1* *yfile1* *xfile2* *yfile2* = *xfile* *yfile*  
Computes a vector difference. Outputs 2 grids: X and Y.
- **dot** *xfile1* *yfile1* *xfile2* *yfile2* = *ofile*  
Computes a dot product.
- **cross** *xfile1* *yfile1* *xfile2* *yfile2* = *xfile* *yfile*  
Computes a vector cross product. Outputs 2 grids: X and Y.
- **lapl** *file1* = *ofile*  
Computes the Laplacian of *file1*.
- **conv** *file1* *file2* [*file3*] = *ofile*  
Computes the convergence of the vector/wind field based on the U wind component in *file1* and the V wind component in *file2*. If *file3* is specified, the convergence is done on the grid field in *file3*.
- **diverg** *file1* *file2* [*file3*] = *ofile*  
Computes the divergence of the vector/wind field based on the U wind component in *file1* and the V wind component in *file2*. If *file3* is specified, the divergence is done on the grid field in *file3*.
- **def1** *file1* *file2* = *ofile*  
Computes the first deformation term of the vector/wind field based on the U wind component in *file1* and the V wind component in *file2*.
- **def2** *file1* *file2* = *ofile*  
Computes the second deformation term of the vector/wind field based on the U wind component in *file1* and the V wind component in *file2*.
- **rvort** *file1* *file2* = *ofile*  
Computes the vorticity or curl of the vector/wind field based on the U wind component in *file1* and the V wind component in *file2*.
- **avort** *file1* *file2* = *ofile*  
Computes the absolute vorticity of the wind field based on the U wind component in *file1* and the V wind component in *file2*.
- **adv** *file1* *file2* *file3* = *ofile*  
Computes the advection of a specified field in *file3* by the wind with the U wind component in *file1* and the V wind component in *file2*.

**NOTE:** The file specifications can be replaced by numeric constants with the syntax:

*#number*

in most of the above functions. For example, the number 5.32 would be entered "**#5.32**".

## EXAMPLES

To get information on the grid:

```
grdmath info uwnd.grd
WXP analysis for 0500Z 21 DEC 96-Surface U wind component (m/s)
Model: 1 Ltype: 1 Level: 0 Var: 33 Hour: 0 Grid: 2
Proj: 1 Lat: 43.00 Lon: -93.00 NX: 25 NY: 17 DX: 1.10 DY: 1.10
Domain: psterio::-93.000:::,43.000,-93.000,25,17,1.1000,1.1000
```

To difference two grids:

```
grdmath temp18.grd sub temp12.grd = diff.grd \
```

```
"6 hour temp change (F)" "12-18Z 12 JUN 96"
```

This subtracts two grids produced by `sfccalc` to produce a 6 hour temperature change. The two quoted expressions change the grid title and date information.

To convert Fahrenheit to Celsius:

```
grdmath copy tempF.grd:C = tempC.grd
```

This just copies the first grid into the second but the units of the first grid are changed after reading so the final grid has the new units.

To find the max and min of a grid:

```
grdmath maxmin uwnd.grd 8.024727 -5.086694
```

To piece two grids together:

```
grdmath piece 2x1 96122012.gmi+24:850:temp:C \
  96122012.gmj+24:850:temp:C = 850_temp.grd
```

This will piece together two aviation grids for Europe and Asia. The "+" in the filename separates the filename from the specification of grid information for parsing. In this case, the 24 hour 850 mb temperature field will be parsed from the multi-grid files. The ":C" specifies to convert the output to Celsius prior to piecing so the resultant pieced grid is in Celsius.

To remap a grid, you just list to grid to remap and the **plot\_domain** specification:

```
grdmath remap 850_temp.grd europe = 850_temp_e.grd
```

With the remap capability, you can difference grids that are not on the same projection such as model grids and those produced from WXP.

## PIPING COMMANDS

This is a powerful concept that allows the output of one WXP command acts as the input to another. You can use either "-" or "**std**" to represent a filename that is either written to standard output or read in from standard input. Here is an example of computing wind speed from component wind fields and then contouring the output:

```
grdmath spd uwnd.grd vwnd.grd = - | contour -ba -
```

You can derive the grids directly from a program like `sfccalc`:

```
( sfccalc -cu=la -re=mw -va=uwnd -pl=none -of=std -me=print; \
  sfccalc -cu=la -re=mw -va=vwnd -pl=none -of=std -me=print ) | \
  grdmath spd - - = - | contour -ba -
```

## FILES

- [Grid files](#)

## SEE ALSO

- [sfccalc](#)
- [upcalc](#)
- [focalc](#)
- [griblook](#)
- [grbcalc](#)

- [grid](#)
- [vector](#)
- [contour](#)

# GRIBLOOK

## NAME

**griblook** - GRIB data query program

## SYNOPSIS

**griblook** [*parameters...*] *filename*

## PARAMETERS

| Command Line                               | Resource                  | Default           | Description   |
|--|---------------------------|-------------------|---|
| <b>-h</b>                                  | <a href="#">help</a>      | No                | Lists basic help information.   |
| <b>-df=<i>filename</i></b>                 | <a href="#">default</a>   | <b>.wxpdef</b>    | Sets the name of the resource file.   |
| <b>-na=<i>name</i></b>                     | <a href="#">name</a>      | <b>griblook</b>   | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                                 | <a href="#">batch</a>     | No                | Run program in batch mode   |
| <b>-me=<i>level</i></b>                    | <a href="#">message</a>   | <b>out2</b>       | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>parameter information for <b>grbcalc</b> - <b>out3</b></li> <li>specific grid information - <b>out4</b></li> <li>low level GRIB information - <b>debug</b></li> </ul>  |
| <b>-fp=<i>filepath</i></b>                 | <a href="#">file_path</a> | current directory | Specifies location of database files.   |
| <b>-dp=<i>datapath</i></b>                 | <a href="#">data_path</a> | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.  |
| <b>-cp=<i>conpath</i></b>                  | <a href="#">con_path</a>  | current directory | Specifies the location (path) of the input converted data files. This may be modified in the name convention file.  |
| <b>-gp=<i>gridpath</i></b>                 | <a href="#">grid_path</a> | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.   |
| <b>-nc=<i>name_conv</i></b>                | <a href="#">name_conv</a> | <b>name_conv</b>  | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=<i>in_file</i></b>                  | <a href="#">in_file</a>   | <b>grib</b>       | Specifies the input file name tag. This value can also be specified with the <b>model</b> resource by looking up the naming convention in the <b>model.lup</b> file. The default is <b>grib</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=<i>out_file</i>[,<i>out...</i>]</b> | <a href="#">out_file</a>  | None              | Specifies what type of grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII grid or raw file</li> <li><b>wxpb</b> - WXP binary grid file</li> <li><b>raw</b> - WXP ASCII raw file</li> <li><b>cdf</b> - netCDF binary grid file</li> </ul> <p>Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type:</p> <pre>wxp:eta_grd</pre> <p>A full filename can be specified which can contain name</p> |

|   |                            |  |   |
|---|----------------------------|--|---|
|   |                            |  | convention tags.  |
| <b>-pf</b> = <i>file_param</i>          | <a href="#">file_param</a> | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul>   |
| <b>-mf</b> = <i>mapfile</i> [,...]      | <a href="#">map_file</a>   | <b>wxp.map</b> or<br><b>cont.dat</b> ,<br><b>country.dat</b> for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.  |
| <b>-cu</b> =[ <i>hour</i>   <b>la</b> ] | <a href="#">current</a>    | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho</b> = <i>hour</i>                | <a href="#">hour</a>       | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-mo</b> = <i>model</i>               | <a href="#">model</a>      | <b>ngm</b> or first<br>model listed in<br><b>model.lup</b> file.                                     | Specifies the model to use as input data. These are defined in the <b>model.lup</b> file. This file can define how model grids are pieced together. Some samples are: <ul style="list-style-type: none"> <li>• <b>ngm</b> -- Nested Grid Model</li> <li>• <b>eta</b> -- ETA model</li> <li>• <b>avn</b> -- Aviation model</li> <li>• <b>mrf</b> -- Medium Range Forecast model</li> </ul>   |
| <b>-gt</b> = <i>grid_type</i>           | <a href="#">grid_type</a>  | None.<br>Displays all grids  | This specifies which type of grid type to specifically parse for. This is a single numeric value which represents the domain that the grid covers. For example, the ETA model is on grid 211.   |
| <b>-gn</b> = <i>grid_num</i>            | <a href="#">grid_num</a>   | None.<br>Displays all grids  | This specifies which type of grid to specifically parse for. The format is as follows:<br><br>+ <i>time</i> : <i>level</i> : <i>var</i> [: <i>units</i> ]<br>or<br># <i>num</i> [: <i>units</i> ]<br><br>The value of <i>num</i> represents the sequence number of the grid in the file. Grids are numbered starting at 0 (zero). The values of <i>time</i> , <i>level</i> , and <i>var</i> make it possible to specify the grid based on typical grid information. Any valid forecast time, level and variable can be specified. For more information, use the full listing with the <b>out3</b> message level. Possible values for each are: <ul style="list-style-type: none"> <li>• <b>time</b>:<br/>init,06,12,18,24,30,36,42,48,60,72,d3,d4,d5,d6,d7,d8,d9,d10</li> <li>• <b>level</b>: sfc, sl (sea level), trop, mwnd, 1000, 850, 700, 500, 400, 300, 200, 100</li> <li>• <b>var</b>: temp, rhum, pres, vvel, uwnd, vwnd, prec, lift</li> </ul> |
| <b>-ou</b> = <i>output</i>              | <a href="#">output</a>     | None.  | This specifies any additional output to be produced by the program:   |

|                                 |                           |  |  |
|---------------------------------|---------------------------|--|--|
|                                 |                           |  | <ul style="list-style-type: none"> <li>• <b>hdrfile</b> - produce a header file for the input grid file</li> </ul>   |
| <b>-pa=param</b><br>[,param...] | <a href="#">parameter</a> | None   | Specifies additional plotting parameters. See the <a href="#">parameter</a> resource for more details. <ul style="list-style-type: none"> <li>• <b>nodhr</b> - Don't use header file for quick lookup</li> <li>• <b>usehdr</b> - Use a header file for quick lookup</li> <li>• <b>max_bad=limit</b> - Specifies limit of bad values that causes rest of GRIB decoding to stop</li> </ul> |
| <b>filename[#seq]</b>           | <a href="#">filename</a>  | None<br>User Prompt<br>Batch: current= <b>la</b> | The name of the grid data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files.  |

## DESCRIPTION

This program scans a multigrid file and displays vital information on each grid contained within the file. This is helpful in determining which grids are available to be contoured by the **grbcalc** program.

The program will display data from many model sources. The model type is specified using the [model](#) resource. The specified string is first looked up in the **model.lup** file which cross-references the file to a file naming convention. If the grids are split for a particular model type, the program will use the first grid name convention in parsing.

The programs starts off by prompting the user for input grid file name. The list of available grid files is based on the file name convention which is dependent on the model type and the file name tag associated with it. The tag can be specifically changed with the [in\\_file](#) resource. The user may specify the input file either via the command line of through the [current](#) resource.

The default output gives general information:

```

0: ETA analysis for 0000Z 29 SEP 97-100 mb Vertical velocity (Pa/s)
1: ETA analysis for 0000Z 29 SEP 97-150 mb Vertical velocity (Pa/s)
2: ETA analysis for 0000Z 29 SEP 97-200 mb Vertical velocity (Pa/s)
3: ETA analysis for 0000Z 29 SEP 97-250 mb Vertical velocity (Pa/s)
4: ETA analysis for 0000Z 29 SEP 97-300 mb Vertical velocity (Pa/s)
...
448: 12 hour ETA valid 12Z MON 29 SEP 97-500 mb Abs vorticity (/s)
449: 12 hour ETA valid 12Z MON 29 SEP 97-250 mb Abs vorticity (/s)
450: 12 hour ETA valid 12Z MON 29 SEP 97-Whole atms Precipitable water (kg/m^2)
451: 12 hour ETA valid 12Z MON 29 SEP 97-0-180 mb AG 4 lyr lifted index (dK)
452: 12 hour ETA valid 12Z MON 29 SEP 97-2 m AG Temperature (K)
453: 12 hour ETA valid 12Z MON 29 SEP 97-2 m AG Relative humidity (%)
...

```

To help in specifying time, level and variable parameters to **grbcalc**, toggle message level **out3**:

```

448: 12 hour ETA valid 12Z MON 29 SEP 97-500 mb Abs vorticity (/s)
Model: ETA Time: 12hr Level: 500mb Variable: avort Units: /s
449: 12 hour ETA valid 12Z MON 29 SEP 97-250 mb Abs vorticity (/s)
Model: ETA Time: 12hr Level: 250mb Variable: avort Units: /s
450: 12 hour ETA valid 12Z MON 29 SEP 97-Whole atms Precipitable water (kg/m^2)
Model: ETA Time: 12hr Level: whole Variable: pwat Units: kg/m^2
451: 12 hour ETA valid 12Z MON 29 SEP 97-0-180 mb AG 4 lyr lifted index (dK)
Model: ETA Time: 12hr Level: 0-180mb_ag Variable: lift4 Units: dK
452: 12 hour ETA valid 12Z MON 29 SEP 97-2 m AG Temperature (K)
Model: ETA Time: 12hr Level: 2m_ag Variable: temp Units: K

```

To get more information on the internal data within the grid, move the message level to **out4**:

```

0: ETA analysis for 0000Z 29 SEP 97-100 mb Vertical velocity (Pa/s)
Model: ETA Time: init Level: 100mb Variable: vvel Units: Pa/s
Model: 89 Ltype: 10 Level: 100 Var: 39 Hour: 0 Grid: 211
Proj: 4 Lat: 40.62 Lon: -100.51 NX: 93 NY: 65 DX: 0.63 DY: 0.63
lamb: -95.000:25.000:25.000:1.277,40.620,-100.510,93,65,0.6263,0.6263

```

This will list internal numbers which reflect data directly from the GRIB product. This can be helpful in determining whether a grid, model or variable that is not currently supported by WXP. In the case of model files with header file, the domain information will not print since only the header file is used to query this information. By going to debug message level, much more information from the GRIB product is available.

### Querying Specific Grids

If a specific grid is needed, the **grid\_num** resource can be specified. Once specified, only the grids that match the parameters are listed in the output. The format of the **grid\_num** resource is as follows:

```
+time:level:var[:units]
or
#num[:units]
```

The value of **num** represents the sequence number of the grid in the file. Grids are numbered starting at 0 (zero).

The values of **time**, **level**, and **var** make it possible to specify the grid based on typical grid information. Any valid forecast time, level and variable can be specified. For more information, use the full listing with the **out3** message level. Possible values for each are:

- **time:** init,06,12,18,24,30,36,42,48,60,72,d3,d4,d5,d6,d7,d8,d9,d10
- **level:** sfc, sl (sea level), trop, mwnd, 1000, 850, 700, 500, 400, 300, 200, 100
- **var:** temp, rhum, pres, vvel, uwnd, vwnd, prec, lift

A value of **miss** can be used in any of the 3 categories if all grids matching the other parameters is needed. For example, if all the 24 hour products are needed, specify the time and leave the other two as **miss**.

If the grid file contains multiple grid types such as 207 and 211 in the case of the ETA model, then the **grid\_type** resource can be specified to obtain the correct grid.

### Output Files

By default, **griblook** does not create any output other than the listings. If an output file is desired, the **out\_file** resource can be specified to create a file to dump the requested grids. It should be noted that this should be used only if a subset of the total grids is needed. The value of the **out\_file** resource is the name convention to be used for the output file.

### Header Files

This program can generate header files if needed. By specifying "**-ou=hdrfile**", the program will search through the model file and generate an appropriate header file. If the header file exists, the program will ask the user to overwrite or append. If the user selects append, the program will start the header search after the last header in the header file. This can significantly reduce runtime.

## FILES

### SEE ALSO

- [grbcalc](#) - the GRIB contouring program
- [contour](#) - the grid contouring program
- [vector](#) - the grid vector plotting program

# GRID

## NAME

**grid** - Raw file gridding and contouring program

## SYNOPSIS

**grid** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description  |
|------------------------------|----------------------------|-------------------|--|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.  |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxpdef           | Sets the name of the resource file.  |
| <b>-na=name</b>              | <a href="#">name</a>       | grid              | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode  |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• gridding information - <b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | raw               | Specifies the input file name tag. The default is <b>raw</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of raw or grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII grid or raw file</li> <li>• <b>wxpb</b> - WXP binary grid file</li> <li>• <b>raw</b> - WXP ASCII raw file</li> <li>• <b>cdf</b> - netCDF binary grid file</li> </ul> Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>raw_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type:<br><br>wxp:sfc_grd |

|   |                              |  |  |
|---|------------------------------|--|--|
|   |                              |  | A full filename can be specified which can contain name convention tags.   |
| <b>-pf=</b> <i>file_param</i>   | <a href="#">file_param</a>   | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul>  |
| <b>-cf=</b> <i>cityfile</i>   | <a href="#">city_file</a>    | <b>sao.cty</b>   | Specifies the name of the city database file used in finding station locations.  |
| <b>-mf=</b> <i>mapfile</i> [,...]   | <a href="#">map_file</a>     | <b>wxp.map</b> or <b>cont.dat</b> , <b>country.dat</b> for domains outside North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]   | <a href="#">current</a>      | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=</b> <i>hour</i>   | <a href="#">hour</a>         | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-id=</b> <i>identifier</i>   | <a href="#">identifier</a>   | None   | Used to plot a specific observation. If not specified, all stations will be plotted.   |
| <b>-pd=</b> <i>domain</i>   | <a href="#">plot_domain</a>  | User prompt<br>Batch: first domain in <b>wxp.reg</b> file.                               | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-gd=</b> <i>grid_domain</i>  | <a href="#">grid_domain</a>  | Same as plot domain  | This specifies the domain to use when fitting data to a grid. If this is not specified, it is the <b>plot_domain</b> .   |
| <b>-re=</b> <i>region</i>   | <a href="#">region</a>       | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-pr=</b> <i>prior</i>  | <a href="#">stat_prior</a>   | Depends on domain size   | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.   |
| <b>-oa=</b> <i>smooth</i><br>[, <i>radinf</i> ][, <i>passes</i> ]<br>[, <i>converg</i> ]<br>[, <i>min_sta</i> ] | <a href="#">object_param</a> | <b>1,3,1,,2,1</b>  | Specifies the objective analysis parameters. There are 5 parameters including smoothing, radius of influence, number of passes, convergence parameter and minimum number of stations.  |
| <b>-pl=</b> <i>plot_type</i>  | <a href="#">plot_type</a>    | User prompt<br>Batch: fill contours or vectors   | Specifies the type of output plot. By default, the user is prompted for the plot type. To produce a contour or vector plot, use the following plot types:<br><b>Contours:</b> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <b>Vectors:</b> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> </ul> |

|  |                              |   |   |
|--|------------------------------|---|---|
|  |                              |   | <ul style="list-style-type: none"> <li>• <b>none</b> -- for no output plot</li> </ul>   |
| <b>-sc=scale_factor</b>                | <a href="#">plot scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>                      | <a href="#">plot format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.  |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>              | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>                 | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.   |
| <b>-in=interval</b>                    | <a href="#">con interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b>            | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.  |
| <b>-cb=base</b>                        | <a href="#">con base</a>     | Lowest contour<br>value, maximum<br>wind speed                          | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.  |
| <b>-fl=font_list</b>                   | <a href="#">font list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>                 | <a href="#">color table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>               | <a href="#">color text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the text color for labels above and below the plot.   |
| <b>-com=color[:attr]</b>               | <a href="#">color map</a>    | <b>red</b>  | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>              | <a href="#">color line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.  |
| <b>-cocm=color[:attr]</b>              | <a href="#">color cmap</a>   | <b>black</b>  | Specifies the color for color fill contour maps.  |
| <b>-coco=color[:attr]</b>              | <a href="#">color cont</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for line contours.  |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color fill</a>   | A predefined<br>color set based<br>on the <b>wxp.clr</b><br>color table | Specifies the color fill contour color set. See the <a href="#">color_fill</a> resource for more details.   |
| <b>-cola=color[:attr]</b>              | <a href="#">color label</a>  | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the line contour label color.   |
| <b>-cocl=color[:attr]</b>              | <a href="#">color clabel</a> | <b>black</b>  | Specifies the color fill contour label color.   |
| <b>-de=device[,...]</b>                | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                                    | Specifies the output device.  |
| <b>-ti=title</b>                       | <a href="#">title</a>        | Program name  | Specifies the window title.   |
| <b>-ic</b>                             | <a href="#">icon</a>         | <b>Off</b>  | Specifies whether to open window in iconified mode.   |
| <b>-bg=color</b>                       | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse<br>plots                       | Specifies the background color of the window.   |
| <b>-ge=geometry</b>                    | <a href="#">geometry</a>     | <b>640x480</b>  | Specifies the window and paper geometry.  |
| <b>filename[-field]</b>                | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: none                                      | The name of the raw file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. The extra information following the "-" defines which field to use within a raw file. The field specification can be a column number or a field identifier. The field identifier is a string that accompanies a column header in the raw file or a string that precedes the data such as HGIRG=5.24 |

## DESCRIPTION

This program grids and contours raw data on a background map. The input to the program is a raw file which can be produced by programs like [sfcwx](#) or [upairwx](#) programs. Then, based on a plot domain, a plot is generated.

The programs starts off by prompting the user for input raw file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **raw** tag but this can be changed with the [in\\_file](#) resource.

In order to get specific information from a raw file, a field specifier can be appended to the filename.

*filename[-field]*

The field specification can be a column number or a field identifier. The field identifier is a string that accompanies a column header in the raw file or a string that precedes the data such as HGIRG=5.24

Next, the program prompts the user for the plot domain. This is done through a region menu which has predefined regions available. If needed a full plot domain can be specified to get a precise domain. The domain can be interactively changed using the mouse once the plot has been generated.

After the region has been set, the city database is read in and a cross-reference table is setup. Since some raw data is not saved with location information, the program will refer to this table to location each station in the input file.

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the converted file is opened and data read in. Any additional computations are performed to produce a derived variable and fit the data to a grid. The background map will plot first unless the output is a color fill contour in which the map plots last. The gridding process uses an objective analysis scheme to produce the grids. The parameters for this process are specified in the **object\_param** resource. Once the grids have been created, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.
- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed.

When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards).

The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

## EXAMPLES

```
grid -re=mw -pl=cf -in=2 -de=d temp.raw
```

This produces a color fill contour plot of temperature for the midwest region at an interval of 2.

```
grid -cu=la -re=mw -of=temp.grd -pl=none temp.raw
```

This will generate grid file **temp.grd**.

## FILES

- **sao.cty** - the surface station location database file

## SEE ALSO

- [mapplt](#) - the raw file plotting program
- [rawmath](#) - the raw file mathematics program
- [contour](#) - the grid file contouring program

# HDRPARSE

## NAME

**hdrparse** - Generates header files from raw ingested data

## SYNOPSIS

**hdrparse** [*parameters...*] *filename*

## PARAMETERS

| Command Line                             | Resource                  | Default   | Description  |
|--|---------------------------|---|--|
| <b>-h</b>                                | <a href="#">help</a>      | No  | Lists basic help information.  |
| <b>-df=filename</b>                      | <a href="#">default</a>   | .wpxdef   | Sets the name of the resource file.  |
| <b>-na=name</b>                          | <a href="#">name</a>      | <b>hdrparse</b>                                 | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                               | <a href="#">batch</a>     | No  | Run program in batch mode  |
| <b>-me=level</b>                         | <a href="#">message</a>   | <b>out2</b>                                     | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>header data - <b>out1</b></li> </ul>  |
| <b>-fp=filepath</b>                      | <a href="#">file_path</a> | current directory                               | Specifies location of database files.  |
| <b>-dp=datapath</b>                      | <a href="#">data_path</a> | current directory                               | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.   |
| <b>-nc=name_conv</b>                     | <a href="#">name_conv</a> | <b>name_conv</b>                                | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>                       | <a href="#">in_file</a>   | <b>raw</b>                                      | Specifies the input file name tag. The default is <b>raw</b> , but it can be modified to any value in the filename convention file to reflect the type of file to parse. A full name convention can be specified as well.                                |
| <b>-cu=[hour la]</b>                     | <a href="#">current</a>   | None  | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=hour</b>                          | <a href="#">hour</a>      | None  | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=num_hour</b>                      | <a href="#">num_hour</a>  | <b>0</b>  | This specifies the number of hours that will be searched for hourly data.  |
| <b>-pa=param</b><br>[, <i>param...</i> ] | <a href="#">parameter</a> | None  | Extra parameters (none currently)  |
| <i>filename</i> [# <i>seq</i> ]          | <a href="#">filename</a>  | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-WXP files.   |

## DESCRIPTION

When parsing data, most of the data is processed a line at a time. This means that searching for a particular product in a file can consume tremendous amounts of time. To simplify the process, there are header files which list the product headers in a separate file along with byte offsets into the actual ingested data file. This file then becomes a lookup table for specific products. The use of a header file speeds up searching for products by nearly an order of magnitude and is recommended for LDM ingested data. Since the LDM does not create header files directly, the **hdrparse** program can be used to post process the LDM data and create the header files.

To run **hdrparse**, the file name convention file must be set up with a header file name syntax:

|         |                 |                 |
|---------|-----------------|-----------------|
| for_dat | %D/%h%m%d%y.for | %D/hd%h%m%d.for |
| sev_dat | %D/%h%m%d%y.sev | %D/hd%h%m%d.sev |
| sum_dat | %D/%h%m%d%y.sum | %D/hd%h%m%d.sum |
| cli_dat | %D/%h%m%d%y.cli | %D/hd%h%m%d.cli |

For each listed file type, there is the name convention for the ingested file plus a second listing which is the name convention of the header file. If the second name convention is omitted, WXP assumes there are no header files for this particular type of file.

Once the name convention file is set up to use header files, run header parse on the type of file:

```
hdrparse -cu=la -if=for_dat
```

and this will generate the header file. The header output will appear on the screen. The **num\_hour** resource can be used to create header files over several hours.

Once done, programs like **parse**, **forecast** and **fouswx** will run faster since the lookup table has been created and quick lookup can occur. This also will minimize network access over an NFS drive since the small header files are used for the bulk of the searching rather than a line by line search through the ingested data file.

**NOTE:** This program does not work for GRIB files. Use [griblook](#) to generate header files for GRIB data.

## FILES

## SEE ALSO

- [parse](#) - generic parsing program
- [forecast](#) - forecast data parsing program

# HEAT

## NAME

**heat** - Heat index computation program

## SYNOPSIS

**heat** [*parameters*] [*temp dewpoint*]

## PARAMETERS

| Command Line         | Resource                  | Default     | Description  |
|----------------------|---------------------------|-------------|--|
| <b>-h</b>            | <a href="#">help</a>      | No          | Lists basic help information.  |
| <b>-df=filename</b>  | <a href="#">default</a>   | .wxpdef     | Sets the name of the resource file.  |
| <b>-na=name</b>      | <a href="#">name</a>      | heat        | Specifies the name used in resource file parsing.  |
| <b>-ba</b>           | <a href="#">batch</a>     | No          | Run program in batch mode  |
| <b>-me=level</b>     | <a href="#">message</a>   | out2        | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• output values - <b>print</b></li> <li>• value labels - <b>mess</b></li> <li>• table - <b>out1</b></li> </ul>  |
| <b>-ta</b>           | <a href="#">table</a>     | none        | Specifies to print out a table. This is the default if temperature and speed aren't prespecified.  |
| <b>-pa=parameter</b> | <a href="#">parameter</a> | none        | Specifies additional parameters: <ul style="list-style-type: none"> <li>• <b>heat</b> - display heat index</li> <li>• <b>thi</b> - display temperature humidity index</li> <li>• <b>dew</b> - use temperature and dewpoint</li> <li>• <b>rh</b> - use temperature and relative humidity</li> </ul> |
| <i>temp</i>          | <a href="#">value</a>     | none        | Specifies the temperature in F.  |
| <i>dewpoint</i>      |                           | User prompt | Specifies the dewpoint in F (use relative humidity when <b>rh</b> parameter set).  |

## DESCRIPTION

The heat index program calculates the heat index from the specified temperature and dewpoint. If *temp* and *dewpoint* are not specified, the program first displays a heat index chart and then prompts the user for a temperature value in F and an associated dewpoint in F. From this information, the heat index is calculated. The heat index is relatively meaningless for temperatures less than 65F, since the skin loses more heat than it produces below this temperature. Also, the heat index is meaningless for dewpoints less than 65F, since the drier air leads to evaporative cooling of the skin's surface. A second index called the temperature-humidity index is calculated for comparison and a textual heat stress condition such as normal, low, medium, high or extreme is displayed. For example:

```
heat 90 70
Temperature (F):  90.0
Dewpoint (F):   70.0
Heat Index (F):  95.4
Temperature-Humidity Index (F):  81.0   High
```

Or to plot a table:

```
heat -table
HEAT INDEX CALCULATION (Ver 5.0B25-WIN32)

Heat Index
```

|                | Dew Point(F) |       |       |       |       |       |       |       |
|----------------|--------------|-------|-------|-------|-------|-------|-------|-------|
|                | 50.0         | 55.0  | 60.0  | 65.0  | 70.0  | 75.0  | 80.0  | 85.0  |
| Temperature(F) |              |       |       |       |       |       |       |       |
| 65:            | 60.8         | 62.0  | 63.3  | 64.8  |       |       |       |       |
| 70:            | 70.5         | 71.4  | 72.5  | 73.8  | 75.3  |       |       |       |
| 75:            | 75.7         | 76.3  | 76.8  | 77.2  | 77.8  | 79.3  |       |       |
| 80:            | 78.3         | 79.2  | 80.2  | 81.4  | 83.0  | 85.5  | 89.5  |       |
| 85:            | 82.0         | 83.2  | 84.7  | 86.6  | 89.1  | 92.5  | 97.6  | 105.2 |
| 90:            | 86.5         | 87.9  | 89.8  | 92.2  | 95.4  | 99.8  | 105.8 | 114.2 |
| 95:            | 91.4         | 93.1  | 95.3  | 98.1  | 101.9 | 107.0 | 113.7 | 122.9 |
| 100:           | 96.6         | 98.5  | 101.0 | 104.2 | 108.4 | 113.9 | 121.2 | 130.9 |
| 105:           | 101.9        | 104.0 | 106.7 | 110.2 | 114.7 | 120.6 | 128.3 | 138.3 |
| 110:           | 107.2        | 109.5 | 112.4 | 116.1 | 120.9 | 127.0 | 134.9 | 145.1 |
| 115:           | 112.4        | 114.8 | 117.9 | 121.8 | 126.8 | 133.1 | 141.2 | 151.4 |
| 120:           | 117.4        | 120.0 | 123.2 | 127.3 | 132.4 | 138.9 | 147.0 | 157.3 |

**FILES****SEE ALSO**

# HPGLCAT

## NAME

**hpglcat** - HPGL concatenation program

## SYNOPSIS

**hpglcat** [*options...*] [*file...*]

## PARAMETERS

| Options                     | Description  |
|-----------------------------|--|
| <b>-h</b>                   | Lists help information.  |
| <b>-port</b>                | Specifies to print plot in portrait mode.                        |
| <b>-land</b>                | Specifies to print in landscape mode.                            |
| <b>-pcl</b>                 | Specifies to add PCL header (default)                            |
| <b>-lj</b>                  | Specifies to rotate page for laserjet printers                   |
| <b>-lwidth</b> <i>width</i> | Specifies the default line width in picas.                       |
| <b>-rot</b> <i>rotation</i> | Specifies the rotation angle for the plot in degrees.            |
| <b>-output</b> <i>file</i>  | Specifies output print file or device (default standard output). |
| <i>file</i>                 | Input HPGL stub  |

## DESCRIPTION

This program adds HPGL header and trailer information onto files to prepare them for printing. This is used when the HPGL append option has been specified with the **device** resource. The HPGL header contains PCL commands to switch into HPGL mode plus initialization commands for the printer. The HPGL stub is now written to the output and then the HPGL trailer which has a page end and the PCL sequence to switch from HPGL back to PCL. Now the file is ready for printing or plotting. Obviously, only one header and trailer can be contained in a particular HPGL file for a WXP image.

If no output filename is given, **hpglcat** reads from standard input. Otherwise, **hpglcat** will read each file from the command line, printing the header before the first file and placing the trailer after the last file. The output of **hpglcat** is standard output unless the **output** parameter is specified or the **wxphp\_out** environment variable is set:

```
setenv wxphp_out HPGL_file
setenv wxphp_out '|program'
```

The output parameter will also recognize the output types along with "-" for output to standard output.

## SEE ALSO

- [pscat](#) - Postscript concatenation program
- [gifcat](#) - GIF file concatenation program

# HURRICANE

## NAME

**hurricane** - Parses hurricane products and reformats for a web page.

## SYNOPSIS

**hurricane** [*parameters...*] *filename*

## PARAMETERS

| Command Line                             | Resource                   | Default            | Description  |
|--|----------------------------|--------------------|--|
| <b>-h</b>                                | <a href="#">help</a>       | No                 | Lists basic help information.  |
| <b>-df=filename</b>                      | <a href="#">default</a>    | .wxpdef            | Sets the name of the resource file.  |
| <b>-na=name</b>                          | <a href="#">name</a>       | hurricane          | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                               | <a href="#">batch</a>      | No                 | Run program in batch mode  |
| <b>-me=level</b>                         | <a href="#">message</a>    | out2               | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>table of storm data - <b>out1</b></li> <li>product level information - <b>out3</b></li> <li>parsed information from product - <b>out4</b></li> </ul>  |
| <b>-fp=filepath</b>                      | <a href="#">file_path</a>  | current directory  | Specifies location of database files.  |
| <b>-dp=datapath</b>                      | <a href="#">data_path</a>  | current directory  | Specifies the location (path) of the input raw tropical data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>                       | <a href="#">con_path</a>   | current directory  | Specifies the location (path) of the output data files. This will be a root directory for hurricane data such as /data/hurricane.  |
| <b>-nc=name_conv</b>                     | <a href="#">name_conv</a>  | name_conv          | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>                       | <a href="#">in_file</a>    | trp_dat            | Specifies the input file name tag. The default is <b>trp_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well. There are two file types that can also be specified: <ul style="list-style-type: none"> <li><b>raw_wp</b> - western Pacific data files</li> <li><b>raw_io</b> - Indian Ocean data files</li> </ul> If either of these is not specified, standard TPC or JTWC format is assumed. |
| <b>-cu=[hour]la</b>                      | <a href="#">current</a>    | None               | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=hour</b>                          | <a href="#">hour</a>       | None               | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=num_hour</b>                      | <a href="#">num_hour</a>   | 0                  | This specifies the number of hours that will be searched for hourly data.  |
| <b>-id=identifier</b>                    | <a href="#">identifier</a> | None               | Used to decode a specific storm (reserved for future use).   |
| <b>-pa=param</b><br>[ <i>,param...</i> ] | <a href="#">parameter</a>  | None               | Extra parameters:  |
| <i>filename</i> [ <i>#seq</i> ]          | <a href="#">filename</a>   | None<br>User Pompt | The name of the raw data file to be parsed. An optional sequence number can be added to designate the time for non-  |

|  |  |                                      |
|--|--|--------------------------------------|
|  |  | Batch: current= <b>la</b> WXP files. |
|--|--|--------------------------------------|

## DESCRIPTION

The hurricane program parses hurricane products in order to generate a wide range of data files containing information on specific hurricanes. The input to the program is raw tropical data file from the TPC (Tropical Prediction Center) and the Joint Typhoon Warning Center (JTWC) which can contain any number of products. To simplify the process, it is highly recommended to save the hurricane products (**WTxx**) in a separate file type as specified by the **trp\_dat** file name tag.

The program parses three types of TPC products:

- **ADVISORY** -- including intermediates and specials and gets wind speed, pressure and location info.
- **FORECAST/ADVISORY** -- these are more regularly formatted and are used to get location, speed and pressure and saves the general information on hurricane for later insertion into the **storm.html** file.
- **DISCUSSION** -- it parses the forecast track information from this product and saves the track lines for insertion into the **storm.html** file.
- **WARNING** - Issued by the JTWC and contains tropical storm and typhoon advisories for the western Pacific.

The output of the program is the following set of files for each active storm:

- **track.dat** - text based track file including forecasted track. This is intended for general use and printing.

```
Date: 24 AUG-04 SEP 1996
Hurricane FRAN
ADV  LAT   LON    TIME    WIND  PR  STAT
  1  14.40  -25.10  24/03Z   30  1007 TROPICAL DEPRESSION
  2  14.80  -26.70  24/09Z   30  1007 TROPICAL DEPRESSION
  3  14.50  -29.20  24/15Z   30  1007 TROPICAL DEPRESSION
...
41A 25.70  -73.10  04/00Z  100   963 HURRICANE-3
+12 26.10  -74.50  04/06Z   95    - HURRICANE-2
+24 27.40  -76.40  04/18Z  100    - HURRICANE-3
+36 29.00  -78.10  05/06Z  105    - HURRICANE-3
+48 30.90  -79.80  05/18Z  110    - HURRICANE-3
+72 35.00  -83.00  06/18Z   40    - TROPICAL STORM
```

- **track.raw** - this is very similar to the track.dat file but it is formatted as a WXP RAW file for use with [mapplt](#) to create tracking charts.
- **track.dom** - the domain for the entire storm track used by mapplt to plot tracking chart. This calculates the range of lat and lon and adds 5 degrees in order to create a plot domain that fully encompasses the entire track. The output is a full WXP plot domain specification that can be used by a program like mapplt to create a tracking chart:

```
mapplt -pd=fi:track.dom track.raw-LINE
```

Example file contents:

```
11,24.70,-54.05,25,17,2.83,2.83
```

- **last.dom** - last position of storm used by xsat for satellite plots. The file is a full WXP plot domain specification so it can be used by xsat directly:

```
xsat ... -pd=fi:last.dom
```

- **storm.html** - full HTML file for that storm, uses image.list file to insert image snapshots of storm, also inserts segments of latest TPC advisories as described above.
- **image.list** - this file is a list of images that have been saved for viewing. This file is NOT created by **hurricane** but is used by the program to create links in the **storm.html** file. The file is a list of image file names along with a description:

```
170903i.gif Infrared image of storm east of Bahamas (17Z Sep 03)
```

- **storm.line** - single line status listing for storm. Example:

```
Hurricane FRAN      24 AUG-04 SEP 100 963 3
```

The program puts these files in directories sorted by storm, region and year. The above files are put in the directories for each storm. The directory structure looks like:

|                 |                  |                 |                  |                   |                  |                 |                  |
|-----------------|------------------|-----------------|------------------|-------------------|------------------|-----------------|------------------|
| <i>storm1</i>   | <i>storm2...</i> | <i>storm1</i>   | <i>storm2...</i> | <i>storm1</i>     | <i>storm2...</i> | <i>storm1</i>   | <i>storm2...</i> |
| <i>year1</i>    |                  | <i>year2...</i> |                  | <i>year1</i>      |                  | <i>year2...</i> |                  |
| <i>region1</i>  |                  |                 |                  | <i>region2...</i> |                  |                 |                  |
| <b>con_path</b> |                  |                 |                  |                   |                  |                 |                  |

The regions include:

- **atlantic** - Atlantic storms
- **e\_pacific** - Eastern Pacific
- **c\_pacific** - Central Pacific
- **w\_pacific** - Western Pacific

Finally, the program produces a listing file which is put in each year directory:

- **storm.list** - list of currently active storm, used for plotting all the data.

This file is only a list of those storms that were active within the period parsed. This list can be used to determine which storms need to be updated on a regular basis. Also, this list can be parsed and new storms added to an overall **names** file which has all the named storms for a particular year.

### Automating the Hurricane Processing

The hurricane program will process each storm that has advisories in the raw tropical files that are in the range specified by **num\_hour** resource. When reprocessing the data, **hurricane** reads in the **track.dat** file for every storm that it sees in the raw files to obtain any older advisory information, adding the new advisories onto the end of the file.

There are a couple of scripts that go with the **hurricane** program. The first is **hur\_update** which runs all the tasks to generate images and HTML pages based on the information generated by **hurricane**. This script should be run in cron once every 2-4 hours or roughly 10-15 minutes after each advisory in order to get the latest advisories. The script does the following tasks:

- it runs **hurricane** to process all the latest hurricane information.
- it uses the **storm.list** file to generate a list of the currently active storms. It updates a **names** file with each new storm.
- it generates satellite images for each active storm using **xsat** with the **last.dom** file. The visible file is **sat\_vis.gif** and the IR file is **sat\_ir.gif**. It also generates a small IR image named **sat\_ir\_s.gif** for inclusion into the **storm.html** file.

- it generates full Atlantic visible, IR and water vapor images on the same projection as the yearly tracking chart.
- it runs the **hur\_plt** script.
  - generates a yearly tracking chart based on the names in the **names** file. When a new storm develops, the **hur\_update** will add its name to this file. You can delete names out of this file when depressions become tropical storms.
  - generates storm based tracking charts using mapplt and the **track.dom** file.
  - it generates small track charts for inclusion into the **storm.html** file.
- it runs **hur\_html\_setup** which is a **Perl** script that generates a large HTML file for each region and year named **index.html**.
  - it inserts each of the **storm.line** files for a general storm summary
  - it generates a HTML table using all the **storm.html** files generated by the hurricane program.

When all this is completed, there is an up to date HTML hurricane page with all the tracking charts updated and current satellite images.

## EXAMPLES

```
hurricane -cu=1a -nh=-6
```

This updates the hurricane files for the last 6 hours. This should be run from cron either by invoking the **hurricane** program or the **hur\_update** script.

## FILES

## SEE ALSO

- **hur\_update** - a C shell script that links all the programs together
- **hur\_plt** - a C shell script that plots hurricane data
- **hur\_html\_setup** - a Perl script that generates index HTML files.

# INGEST

## NAME

**ingest** - The data ingest and selection program

## SYNOPSIS

**ingest** [*parameters...*] *filename*

## PARAMETERS

| Command Line                    | Resource                  | Default           | Description  |
|---------------------------------|---------------------------|-------------------|--|
| <b>-h</b>                       | <a href="#">help</a>      | No                | Lists basic help information.  |
| <b>-df=filename</b>             | <a href="#">default</a>   | .wxpdef           | Sets the name of the resource file.  |
| <b>-na=name</b>                 | <a href="#">name</a>      | ingest            | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                      | <a href="#">batch</a>     | No                | Run program in batch mode  |
| <b>-me=level</b>                | <a href="#">message</a>   | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• product headers - <b>out1</b></li> <li>• product descriptions - <b>out2a</b></li> <li>• output filenames - <b>out2c</b></li> <li>• product contents - <b>out3</b></li> </ul>  |
| <b>-fp=filepath</b>             | <a href="#">file_path</a> | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>             | <a href="#">data_path</a> | current directory | Specifies location of ingested data files. This is assumed to be the output of the ingest process.   |
| <b>-cp=conpath</b>              | <a href="#">con_path</a>  | current directory |  |
| <b>-if=input</b>                | <a href="#">in_file</a>   | none              | Specifies the type of input to the program: <ul style="list-style-type: none"> <li>• <b>dds</b> - the domestic data feed</li> <li>• <b>pps</b> - the public products data feed</li> <li>• <b>ddp</b> - the domestic plus data feed</li> <li>• <b>ids</b> - the international data feed</li> <li>• <b>604</b> - the FAA 604 data feed</li> <li>• <b>hds</b> - the high res data feed</li> <li>• <b>wxp</b> - WXP ingested data files</li> <li>• <b>kav</b> - Kavouras data files</li> </ul> <p>You can also specify baud rate and parity if needed by adding them after the input type, separated by commas as in:</p> <p>ddp,9600,even</p> |
| <b>-bf=bull_file</b>            | <a href="#">bull_file</a> | ingest.bul        | Specifies the bulletin file. This file contains a list of which products are to be save, the action to be performed and the file naming convention to use.   |
| <b>-lf=log_file</b>             | <a href="#">log_file</a>  | ingest.log        | Specifies the log file for the ingest process. This log file will contain critical information about the status of the ingest program. Messages in the ingest file are all date/time stamped.  |
| <b>-pa=param</b><br>[,param...] | <a href="#">parameter</a> | None              | Extra parameters: <ul style="list-style-type: none"> <li>• <b>cntrl</b> - print control characters as [XX] rather than stripping them for output to standard output.</li> <li>• <b>log_unk</b> - log unknown or unselected products.</li> </ul>  |
| <i>filename</i>                 | <a href="#">filename</a>  | Standard Input    | This is the input filename. This can be either: <ul style="list-style-type: none"> <li>• <b>file</b> - a standard file which needs to be reparsed</li> <li>• <b>fifo</b> - this is a named pipe created with the <b>mknod</b></li> </ul>   |

|  |  |  |  |
|--|--|--|--|
|  |  |  | command. <ul style="list-style-type: none"> <li>• <b>device</b> - this is a character device like a serial port which is used to ingest from NWS data feeds.</li> <li>• <b>socket</b> - this is a network connection using a TCP/STREAM connection. Specify "<b>sock:port</b>" with the port address.</li> </ul> |
|--|--|--|--|

## DESCRIPTION

The ingest is set up to read in and process each of the Family of Service and NOAAPORT feeds. Considering that several megabytes of data are broadcast on each of these feeds each day, the ingest program must offer a means to select products (or discard unneeded ones) and file them in a fashion that makes it easier for programs to search for appropriate data.

The ingest program can receive data from four sources:

- **File** -- this is a file of raw ingested data from FOS or NOAAPORT. This can be fed through the ingest program for addition product selection and management. To specify a file, list the filename on the command line to the ingest program.
- **Serial Port** -- this is a standard RS/232 type serial port which is configured for baud rate and parity. WXP has several presets for various FOS feeds such as domestic data and public products. Otherwise, the port parameters are set with the **in\_file** resource. To specify a serial port, list the device driver (/dev/ttya) or port (COM1 for Windows).
- **Named Pipe (FIFO)** -- this is a named pipe (Unix only). This is a file on disk that acts as a queue where one process can write data to the pipe and the ingest program can read that data from the pipe. This is handy for interfacing the WXP ingestor with non-WXP ingestors. To specify a named pipe, list the filename of the named pipe. WXP will determine if it is a named pipe or a file.
- **Socket** -- a socket is a network connection that acts like a queue. One program feeds data to a socket while the WXP ingestor reads data from the socket and processes it. WXP uses a TCP/STREAM socket to preserve data integrity. The WXP ingestor acts as the socket server and binds itself to the socket. To specify a socket, use the keyword "**sock:port**" with the port address. A recommended port address is something in the range of 5000 (this is to eliminate conflicts with other TCP/IP applications). The other application which acts as a client must know the IP address of the machine the WXP ingestor is running on and the port number it bound to.

The ingest program uses a pattern matching scheme to select products. Each pattern has an associated action that is to be performed on the matched product. These actions include:

- **write** - write the product to a file. If the ingestor matches a new product, the new product will overwrite the contents of the file.
- **append** - append each new matched product to the end of the file. This is the most common action as it is easier to a single file with a few hundred products than it is to manage hundreds of small files.
- **pipe** - pipe the contents of the product to the standard input of a specified program. With the pipe action, further processing of the data can be done before writing the output to file. Also, this can be used to mail products to other users.
- **run** - run a specified program once the matched product is received. This can be used to flag a user when a severe weather statement is received.

## Bulletin File

The ingest programs uses a bulletin file to set up which products are to be selected from the data feed and which actions to perform on them. The bulletin filename is specified with the [bull\\_file](#) resource. The bulletin file contains a list of headers, actions and commands to be performed:

```
header [action] [command/filename...] [header file]
header [action] [command/filename...] [header file]
...
```

The header can specify the exact header or a pattern to which headers can be matched. The headers listed in the file can use the following wildcard characters:

|                         |   |
|-------------------------|---|
| . or ?                  | match a single character                      |
| - or *                  | match any character                           |
| [ <i>letters</i> ]      | match a character from the set.               |
| [ <i>^letters</i> ]     | match any character except those from the set |
| ( <i>str1 str2...</i> ) | match strings                                 |
| _                       | underscore matches a space                    |
| / <i>data</i>           | match extra information                       |

Some example header strings are:

|            |  |
|------------|--|
| AB         | Anything that starts with AB                         |
| S[AP]      | SA or SP   |
| (W AC RG)  | Starts with W or AC or RG                            |
| F[^O]      | Anything that starts with F, second character NOT O  |
| FQUS1_KIND | Full header specification with spaces as underscores |
| *_KIND     | Wildcard match on any product that ends with KIND    |

When the product is GRIB, the header is parsed for specific product parameters. This information can then be used to select the product. The syntax for this selection is:

```
/ [Xvvv] [Xvvv] [Xvvv] . . .
```

Where X is:

**M** -- model number  
**G** -- grid number  
**L** -- level type  
**H** -- level value  
**T** -- forecast time  
**V** -- variable number  
vvv -- the value of the parameter

The values for each parameter are listed in the [WXP Product Description Appendix](#). Using the internal GRIB parameters is more reliable than selecting by the WMO header because more than one product may have the same header:

```
HVAC98 KWBC 070000 from Sea Wave model
HVAC99 KWBC 070000 from Aviation model
```

To separate the two products, use the model specifications: **/M77** for the Aviation model and **/M10** for the Sea Wave model.

## Actions

The actions are:

|               |   |
|---------------|---|
| >>            | append to file with header                                  |
| <b>append</b> | same as above   |
| >             | write to file with header, previous content overwritten     |
| <b>write</b>  | same as above   |
| #             | write to file without header, previous contents overwritten |
| <b>file</b>   | same as above   |
|               | pipe product to listed command                              |
| <b>pipe</b>   | same as above   |
| @             | run command when product complete                           |

|            |               |
|------------|---------------|
| <b>run</b> | same as above |
|------------|---------------|

Also, the action can be prepended by a set of flags:

- **R** -- specifies to save the file as a raw file and not strip control characters.
- **B** -- specifies a product to be a binary product and not strip unprintable characters
- **P** -- specifies to send a PAN message at the completion of a product

### Command or Filename

The command is generally the file to place the output or the command to run with the pipe or run actions. The command can have several escape characters:

#### Examples based on system time 1455Z Jan 12, 1997, product header FPUS5 KIND 281512

| Wildcard | Explanation                         | Example |
|----------|-------------------------------------|---------|
| @tag     | Name convention tag                 |         |
| %Y       | current system year                 | 1997    |
| %y       | current system year (last 2 digits) | 97      |
| %m       | current system month                | 01      |
| %d       | current system day                  | 12      |
| %j       | current system Julian day           | 12      |
| %h       | current system hour                 | 14      |
| %n       | current system minute               | 55      |
| %pd      | product day                         | 28      |
| %ph      | product hour                        | 15      |
| %pn      | product minute                      | 12      |
| %T       | product type                        | FPUS5   |
| %t       | product type (lower case)           | fpus5   |
| %L       | product locale                      | KIND    |
| %l       | product locale (lower case)         | kind    |
| %D       | data_path resource                  |         |
| %C       | con_path resource                   |         |
| %R       | raw_path resource                   |         |
| %G       | grid_path resource                  |         |
| %W       | watch_path resource                 |         |
| %I       | image_path resource                 |         |
| %F       | file_path resource                  |         |

Some of the above wildcards can be preceded with a number. For dates, the number is a modifier which rounds down to the nearest value which is a multiple of that number. For example, "%6h" would round down to the nearest 6 hour boundary. For the previous example, it results in the value 12.

For the product type and locale, this number is used in a substring operation. The first digit of the number is the offset into the string and the second digit refers to the number of characters to use. For example, "%12T" results in "FP". To get "IND", use "%23L".

### Header Files

To aid in the parsing of products from the various feeds, a header file can be created by the ingest program. This essentially lists the header of each product in the file along with its byte offset into the file. Since most parsing is based on header, it is far easier to search the smaller header file than to parse through the much larger product file.

To produce these files automatically by the ingestor, add the file name convention to the end of the line in the bulletin file:

```
F[^O]          >>    %D/%y%m%d%6h_for.wmo  %D/%y%m%d%6h_for.hdr
```

The first name convention listed "%D/%y%m%d%6h\_for.wmo" is the filename where the actual product is saved. The second name convention "%D/%y%m%d%6h\_for.hdr" is where the header file information is saved. The syntax of the file is as follows:

```
offset header / extra
offset header / extra
....
```

where:

- offset -- is the byte offset into the file,
- header -- is the product header in its entirety is listed after the offset
- extra -- extra information about the product which is normally the AWIPS header

A sample from a forecast data header file:

```
0 FPUS86 KPQR 282359 / OPUPDX
3264 FPUS85 KGGW 290001 / OPUGGW
3548 FPAK11 PAYA 282207 / &ZCZC JNULFPYAK
4190 FPUS73 KFGF 282359 / NOWFAR
```

For more information on header files, see the section on [header files](#).

| # Pattern                    | Action     | Filename                                 | Header Filename       |
|------------------------------|------------|--|-----------------------|
| #                            |            |  |                       |
| S[AP]                        | >>-15      | %D/%y%m%d%h_sao.wmo                      |                       |
| S[IMNS]                      | >>-05      | %D/%y%m%d%h_syn.wmo                      |                       |
| SD                           | >>+07      | %D/%y%m%d%h_rad.wmo                      |                       |
| U[^AB]                       | >>-65      | %D/%y%m%d%12h_upa.wmo                    |                       |
| ASUS1_                       | >>         | %D/%y%m%d%3h_frt.wmo                     |                       |
| WWUS40                       | >>         | %D/%y%m%d%6h_wws.wmo                     |                       |
| FO                           | >>         | %D/%y%m%d%12h_mod.wmo                    | %D/%y%m%d%12h_mod.hdr |
| A                            | >>         | %D/%y%m%d%6h_sum.wmo                     | %D/%y%m%d%6h_sum.hdr  |
| C                            | >>         | %D/%y%m%d%6h_cli.wmo                     | %D/%y%m%d%6h_cli.hdr  |
| W                            | >>         | %D/%y%m%d%6h_sev.wmo                     | %D/%y%m%d%6h_sev.hdr  |
| #                            |            |  |                       |
| # Specific forecast products |            |  |                       |
| #                            |            |  |                       |
| FXUS01                       | >          | %D/fore/48hr                             |                       |
| FXUS02                       | >          | %D/fore/3-5d_Hem                         |                       |
| FPUS53_KIND                  |            | /usr/local/bin/parse - -ph=FPUS53_KIND - |                       |
| id=%INZ029                   | -pa=dollar | -of=%D/fore/laf_zone                     | -me=none              |
| *_KIND                       | >>         | %D/Indy/%m%d.dat                         |                       |
| #                            |            |  |                       |
| # HDS products               |            |  |                       |
| #                            |            |  |                       |
| Y/M89                        | >>         | %D/%y%m%d%12h_eta.grb                    | %D/%y%m%d%12h_eta.hdr |
| Y/M39G211                    | >>         | %D/%y%m%d%12h_ngm.grb                    | %D/%y%m%d%12h_ngm.hdr |
| Y/M64G211                    | >>         | %D/%y%m%d%12h_ngm.grb                    | %D/%y%m%d%12h_ngm.hdr |

## Program Output

The default output of the ingest program is to reformat the products, removing the control character sequence and formatting the header and product as follow:

```
** header ***
```

```

product
** header ***
....

```

This allows the ingestor to reparse data ingested by the WXP ingestor to increase granularity of data files. For example, you may want to take the forecast files from the initial ingest and parse for products out of KIND.

When the ingest program is running, it will display a list of the products being broadcast on the data feed. The selected product's header will be preceded by "\*\*\*" and the discarded products will be preceded by "--". The action and the output file will also be displayed.

```

**SAAK70 KAWN 080800 RTD** 97 JAN 8 08:38:29Z
Append to: /usr/wxp/data/97010808.sao
**SACN85 CWA0 080834 ** 97 JAN 8 08:38:29Z
Append to: /usr/wxp/data/97010808.sao
**SPUS70 KWBC 080837 ** 97 JAN 8 08:38:29Z
Append to: /usr/wxp/data/97010808.sao
**SPUS80 KWBC 080837 ** 97 JAN 8 08:38:30Z
Append to: /usr/wxp/data/97010808.sao
**SPCN46 CWA0 080835 ** 97 JAN 8 08:38:30Z
Append to: /usr/wxp/data/97010808.sao
**SACN85 CWA0 080834 ** 97 JAN 8 08:38:30Z
Append to: /usr/wxp/data/97010808.sao
**SXUS91 KNKA 080837 ** 97 JAN 8 08:38:30Z
Append to: /usr/wxp/data/97010808.sfc
**SPCN42 CWA0 080836 ** 97 JAN 8 08:38:30Z
Append to: /usr/wxp/data/97010808.sao
**FPUS3 KBUF 080836 ** 97 JAN 8 08:38:30Z
Append to: /usr/wxp/data/97010806.for
**FPUS4 KBUF 080837 ** 97 JAN 8 08:38:30Z
Append to: /usr/wxp/data/97010806.for

```

If the product contains GRIB data, the GRIB header is decoded to give further information about the product:

```

**HVKA99 KWBC 061200 ** 97 JAN 6 18:58:42Z
AVN analysis - 1000 mb V wind component (m/s)
Append to: /usr/wxp/data/97010612_avnlw.grb
**HVLA99 KWBC 061200 ** 97 JAN 6 18:58:44Z
AVN analysis - 1000 mb V wind component (m/s)
Append to: /usr/wxp/data/97010612_avn0w.grb
**HVMA99 KWBC 061200 ** 97 JAN 6 18:58:47Z
AVN analysis - 1000 mb V wind component (m/s)
Append to: /usr/wxp/data/97010612_avs0e.grb
**HVNA99 KWBC 061200 ** 97 JAN 6 18:58:49Z
AVN analysis - 1000 mb V wind component (m/s)
Append to: /usr/wxp/data/97010612_avs1e.grb
**HVOA99 KWBC 061200 ** 97 JAN 6 18:58:51Z
AVN analysis - 1000 mb V wind component (m/s)
Append to: /usr/wxp/data/97010612_avslw.grb
**HVPA99 KWBC 061200 ** 97 JAN 6 18:58:53Z
AVN analysis - 1000 mb V wind component (m/s)
Append to: /usr/wxp/data/97010612_avs0w.grb
**HPIA98 KWBC 061200 ** 97 JAN 6 18:58:55Z
AVN analysis - Surface Pressure (Pa)
Append to: /usr/wxp/data/97010612_avn0e.grb

```

## Output Files

The ingest program reformats the products when it saves them to file. First it strips the bulk of the control characters out of the file. This is to allow text editors and word processors to be able to read in and process the data. In replacing the control characters, the ingest program delimits headers with asterisks "\*\*\*".

```
** header ***
product
** header ***
....
```

A sample of a DD+ output file is:

```
** FPUS73 KFGF 282359 ***
NOWFAR
```

```
SHORT TERM FORECAST
NATIONAL WEATHER SERVICE EASTERN ND/GRAND FORKS ND
656 PM CDT THU MAY 28 1998
```

```
NDZ006>008-014>016-290600-
BENSON-CAVALIER-PEMBINA-RAMSEY-TOWNER-WALSH-
INCLUDING THE CITIES OF -CAVALIER-DEVILS LAKE-GRAFTON-LANGDON-
656 PM CDT THU MAY 28 1998
```

```
.NOW...
SCATTERED SHOWERS AND AN ISOLATED THUNDERSTORM CAN BE EXPECTED NORTH OF
A LINE FROM CANDO TO GRAFTON THROUGH SUNSET. THE HEAVIER SHOWERS MAY
PRODUCE UP TO ONE HALF AN INCH OF RAIN. WEST WINDS GUSTING TO 25 MPH
WILL DECREASE AFTER SUNSET. BY MIDNIGHT TEMPERATURES WILL RANGE FROM 55
IN CANDO AND PEMBINA TO 63 IN DEVILS LAKE AND GRAFTON.
```

\$\$

```
** FPUS73 KDMX 290003 ***
NOWDSM
```

```
SHORT TERM FORECAST
NATIONAL WEATHER SERVICE DES MOINES IA
703 PM CDT THU MAY 28 1998
```

```
IAZ004>007-015>017-023>028-033>039-290603-
ALGONA-ESTHERVILLE-FORT DODGE-IOWA FALLS-MASON CITY-WATERLOO-
703 PM CDT THU MAY 28 1998
```

```
.NOW...
...A TORNADO WATCH REMAINS IN EFFECT UNTIL 900 M...
EXPECT LITTLE CHANGE IN THE WEATHER EARLY THIS EVENING WITH
PERIODIC SHOWERS AND THUNDERSTORMS. SOME STORMS WILL BE SEVERE WITH
DAMAGING WINDS...LARGE HAIL AND POSSIBLY A TORNADO. BE PREPARED TO
SEEK SAFE SHELTER ON SHORT NOTICE. TEMPERATURES SHOULD MAINLY BE IN
THE 70S WITH COULD BE A BIT COOLER NEAR STORMS.
```

\$\$

```
** FPUS74 KFWD 290004 ***
NOWFTW
```

...

## PAN (Product Arrival Notices) Messages

Product arrival notices are sent at the completion of a product to a specified PAN receiving program. The PAN receiver will use this message to trigger an action based on the arrival of that product. For example, a PAN receiver might be interested in the arrival of severe thunderstorm warning messages so it can warn the user. The PAN message is broadcast over a socket using a UDP transmission. This is a connectionless process where the PAN is sent to a specific address and port and it is up to the PAN receiver to be active and waiting for the message using a receive from call.

The PAN message is sent as a single line of information for each product received by the ingestor. The information in the PAN message is broken up into fields delimited by a bar "|":

```
ID|Server|###|YYYYMMDDhhmmss|WMO/Extra|Filename|Offset|Size
```

Fields:

- ID -- Message Type ID (901 for NOAAPORT)
- Server -- NOAAPORT server number which uniquely identifies server (0-99)
- ### -- Sequence number from server (0-999). Increments by one for each PAN sent from that server. It cycles through numbers 0 to 999 and back to 0.
- YYYYMMDDhhmmss -- Timestamp of when product is sent from WXP ingestor.
- WMO/Extra -- WMO Header plus additional information. For text products, this is the first 20 bytes of the product (newlines and unprintables changed to spaces). This will often contain the AWIPS header. For GRIB products, this is the decoded header information from the GRIB Product Definition Block. Included is the model number (PDS Octet 6), grid type (PDS Octet 7), valid/forecast time (derived from PDS Octet 19,20 and 21), vertical level type (PDS Octet 10), level value (PDS Octet 11-12) and parameter (PDS Octet 9).
- Filename -- Filename including full path. This is the filename that the WXP ingestor saved the product to. **NOTE:** This filename and path may be different from the filename and path you need to access the data. If the data is mounted on an NFS drive, the appropriate NFS path will need to be substituted for the path listed here.
- Offset -- Byte offset of product header in file. This is the exact location (first byte in file is 0) of the start of the product header. An fseek using this number is all that is needed to locate the product.
- Size -- Size in bytes of product from header to end of product including any leading or trailing blank lines

Examples:

```
901|45|909|19980428152512|SDXX99 KWBC 281522 / CMFWS /home/wxp/data/98042815_rad.wmo|75975|2410|
```

- 901 - identifies NOAAPORT PAN message
- 45 - identifies local NOAAPORT server
- 909 - is the sequence number
- 19980428152512 - Date product arrived on server and PAN message sent (depends on server time). It arrived at 15:25:12Z on 24 APR 1998
- SDXX99 KWBC 281522 - WMO header RCMFWS - AWIPS header
- /home/unisys/wxp/data/98042815\_rad.wmo - server filename where product is located. Each file can contain more than one product
- 75975 - byte offset in file
- 2410 - product size in bytes

```
901|45|907|19980428152510|YSRG98 KWBD 281200 PAA / 89 212 4030036 1 0  
66|/home/wxp/model/98042812_eta2.grb|3412593|32943|
```

- 901
- 45 - identifies local NOAAPORT server
- 907 - is the sequence number
- 19980428152510 - Date product arrived on server and PAN message sent

- YSRG98 KWBD 281200 PAA - WMO header 89 212 4030036 1 0 66 - Extra GRIB info, model 89 is Eta model, grid 212 is AWIPS grid 212, time 4030036 is a 6 hour accumulation from forecast hour 30 to 36, level type 1 is surface, level 0 is ignored for surface and parameter 66 is snow depth.
- /home/unisys/wxp/model/98042812\_eta2.grb
- 3412593 - byte offset in file
- 32943 - product size in bytes

## PAN Message Setup

To set up the WXP ingestor for PAN messages the following pieces of information must be added to the "ingest.bul" file. At some point in the file, a PAN configuration line must be added.

```
# PAN Setup
@PAN id=45 sock:steve:5566 sock:dev5:5000 pan.log
```

The "@PAN" is a keyword in the bulletin file for the PAN configuration line. The "id=45" specifies the NOAAPORT unique server ID which is broadcast as field 2 in the PAN message. The rest of the line lists destinations. The "sock" keyword specifies the PAN go over a UDP socket. The string "steve:5566" is the network name of the destination computer and the TCP/IP port number. If the sock keyword is omitted, the PAN is save to the listed filename such as "pan.log". Up to 10 destinations can be listed. Each destination is addressed starting with 0 and going to 9 in the order listed on the PAN line.

By default, no PAN messages are sent even if the PAN line is added to the bulletin file. To enable PAN messages, the "P" flag must be added to the action for each product being saved on the server. For example a product line would look like:

```
# Pattern Action      Filename                      Header Filename
FT                >>    %D/%y%m%d%h_term.wmo %D/%y%m%d%h_term.hdr
```

To enable this product type for PAN messages, add the "P" flag to the action.

```
FT                P>>    %D/%y%m%d%h_term.wmo %D/%y%m%d%h_term.hdr
```

This will send a PAN message to all listed destinations whenever this products is received. If you don't want to send a PAN to all destinations, the destination IDs can be listed:

```
FT                P035>>    %D/%y%m%d%h_term.wmo %D/%y%m%d%h_term.hdr
```

In this case, PAN messages will only be sent to the 0, 3 and 5th destinations.

## Log Files

The ingest program logs appropriate information in a log file. By default, this file is named "ingest.log" and is put in the **file\_path** directory. The program logs when ingest starts and stops, lists all unselected products and notes any corrupted products from HRS. Each entry is timestamped:

```
98 MAY 15 15:11:51Z : Unselected product: GPNG98 KWBC 151200 / GRID 07092 10101
98 MAY 15 15:11:51Z : Unselected product: GPNI98 KWBC 151200 / GRID 07092 10101
98 MAY 15 15:13:18Z : Unselected product: NWUS43 KFSD VERIFY / WVMFSD
98 MAY 15 15:13:20Z : Unselected product: NWUS43 KFSD VERIFY / WVMFSD
98 MAY 15 15:13:20Z : Unselected product: NWUS43 KFSD VERIFY / WVMFSD
```

The log file name can contain name convention wildcard characters such as "/usr/wxp/logs/noaa-%m%d.log" where the %m and %d are replaced with the month and day so that log files are generated for each day the ingestor is running.

## Terminating Ingest

Ingest may be stopped in two ways. First, if the ingest program is running in the foreground, the break or interrupt key may be hit and the message "**Break: do you want to quit (k/y/n):** " appears. This allows the user to quit or return to ingest if the break key was hit by accident. If **y** is specified, the ingest program ends following the end of the current product. If **k** is specified, the ingest program ends immediately. If the ingest program is running as a background task (UNIX only), the user may also issue the **kill** command from the operating system specifying the process identifier of the ingest program.

**OPERATIONS NOTE:** The ingest program may be listed in the `"/etc/rc"` (Unix startup script) or `"autoexec.bat"` (for MS-Windows) so ingest will be started whenever the system is first booted up or powered on. Since no environment variables are set upon system initialization, program resources must be specified by either specifying the resource file with `"-df=/usr/wxp/etc"` or by specifying the `data_path` and `file_path` parameters, respectively .

## FILES

- [ingest.bul](#) - Bulletin file which specifies which products are to be saved. This can be specified with the `bull_file` resource.
- `ingest.log` - Log file records important information from the ingest process. This

## SEE ALSO

- [parse](#)
- [forecast](#)

# LIGHT

## NAME

**light** - NLDN lightning display program

## SYNOPSIS

**light** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default  | Description  |
|------------------------------|----------------------------|--|--|
| <b>-h</b>                    | <a href="#">help</a>       | No   | Lists basic help information.  |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxddef  | Sets the name of the resource file.  |
| <b>-na=name</b>              | <a href="#">name</a>       | <b>light</b>   | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                   | <a href="#">batch</a>      | No   | Run program in batch mode  |
| <b>-me=level</b>             | <a href="#">message</a>    | <b>out2</b>  | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>   |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory  | Specifies location of database files.  |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory  | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory  | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory  | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory  | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory  | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | <b>name_conv</b>   | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-inp=input</b>            | <a href="#">input</a>      | <b>albany</b>  | Specifies the input file type. The default is <b>albany</b> . Possible values are: <ul style="list-style-type: none"> <li><b>albany</b> -- Albany format</li> <li><b>unisis</b> -- Unisis format</li> </ul>  |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | <b>nldn</b>  | Specifies the input file name tag. The default is <b>nldn</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None   | Specifies what type of output file. There is no output file created by the program but GIF files can be created.   |
| <b>-mf=mapfile[,...]</b>     | <a href="#">map_file</a>   | <b>wxp.map</b> or <b>cont.dat</b> , <b>country.dat</b> for domains outside North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu=[hour]la</b>          | <a href="#">current</a>    | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=hour</b>              | <a href="#">hour</a>       | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |

|   |                              |  |  |
|---|------------------------------|--|--|
| <b>-nh=</b> <i>num_hour</i>                         | <a href="#">num_hour</a>     | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.  |
| <b>-pd=</b> <i>domain</i>                           | <a href="#">plot_domain</a>  | User prompt<br>Batch: first domain in <b>wxp.reg</b> file. | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-re=</b> <i>region</i>                           | <a href="#">region</a>       | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-va=</b> <i>variable</i>                         | <a href="#">variable</a>     | <b>intens</b>  | This specifies the variable to use to color the strike points: <ul style="list-style-type: none"> <li>• <b>intens</b> - intensity of strike</li> <li>• <b>time</b> - the minute of the strike</li> <li>• <b>flash</b> - the number of flashes</li> </ul> |
| <b>-sc=</b> <i>scale_factor</i>                     | <a href="#">plot_scale</a>   | <b>1.0</b>   | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm=</b> <i>format</i>                           | <a href="#">plot_format</a>  | None   | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp=</b> <i>param</i><br>[, <i>param</i> ...]    | <a href="#">plot_param</a>   | None   | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.  |
| <b>-dr=</b> <i>drawable</i> [,...]                  | <a href="#">draw</a>         | <b>all</b>   | Specifies which part of the plot is to be drawn.   |
| <b>-la=</b> <i>label</i> [,...]                     | <a href="#">label</a>        | Program defaults   | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-in=</b> <i>interval</i>                         | <a href="#">con_interval</a> | None for plots,<br>User prompt for grids. Batch: <b>0</b>  | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.   |
| <b>-cb=</b> <i>base</i>                             | <a href="#">con_base</a>     | Lowest contour value, maximum wind speed                   | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <b>con_base</b> resource.  |
| <b>-fl=</b> <i>font_list</i>                        | <a href="#">font_list</a>    | <b>modern.fnt</b>  | Specifies a list of fonts to use in the plots.   |
| <b>-ct=</b> <i>color_table</i>                      | <a href="#">color_table</a>  | <b>wxp.clr</b>   | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |
| <b>-cot=</b> <i>color</i> [: <i>attr</i> ]          | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse plots.            | Specifies the text color for labels above and below the plot.  |
| <b>-cod=</b> <i>color</i> [: <i>attr</i> ]          | <a href="#">color_data</a>   | <b>green</b>   | Specifies the color for plotted data.  |
| <b>-com=</b> <i>color</i> [: <i>attr</i> ]          | <a href="#">color_map</a>    | <b>red</b>   | Specifies the color for background maps.   |
| <b>-coln=</b> <i>color</i> [: <i>attr</i> ]         | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>                                      | Specifies the color for lat/lon lines.   |
| <b>-cowl=</b> <i>color</i> [: <i>attr</i> ]         | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse plots.            | Specifies the color for wind barbs.  |
| <b>-cocd=</b> <i>color</i> [: <i>attr</i> ]         | <a href="#">color_cloud</a>  | <b>white</b><br><b>black</b> for inverse plots.            | Specifies the color for cloud cover symbols.   |
| <b>-cowx=</b> <i>color</i> [: <i>attr</i> ]         | <a href="#">color_wx</a>     | <b>yellow</b><br><b>brown</b> for inverse plots            | Specifies the color for weather symbols  |
| <b>-cofr=</b> <i>color</i> [: <i>attr</i> ],<br>... | <a href="#">color_front</a>  | A predefined color set                                     | Specifies the colors for fronts and pressure systems.  |
| <b>-cocm=</b> <i>color</i> [: <i>attr</i> ]         | <a href="#">color_cmap</a>   | <b>black</b>   | Specifies the color for color fill contour maps.   |
| <b>-coco=</b> <i>color</i> [: <i>attr</i> ]         | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse                   | Specifies the color for line contours.   |

|                                     |                              |  |  |
|-------------------------------------|------------------------------|--|--|
|                                     |                              | plots.   |  |
| <b>-cof</b> =[val:]color[:attr],... | <a href="#">color fill</a>   | A predefined color set based on the <b>wxp.clr</b> color table | Specifies the color fill contour color set. See the <b>color_fill</b> resource for more details.   |
| <b>-cola</b> =color[:attr]          | <a href="#">color label</a>  | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the line contour label color.  |
| <b>-cocl</b> =color[:attr]          | <a href="#">color clabel</a> | <b>black</b>   | Specifies the color fill contour label color.  |
| <b>-de</b> =device[,...]            | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                           | Specifies the output device.   |
| <b>-ti</b> =title                   | <a href="#">title</a>        | Program name   | Specifies the window title.  |
| <b>-ic</b>                          | <a href="#">icon</a>         | <b>Off</b>   | Specifies whether to open window in iconified mode.  |
| <b>-bg</b> =color                   | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots                 | Specifies the background color of the window.  |
| <b>-ge</b> =geometry                | <a href="#">geometry</a>     | <b>640x480</b>   | Specifies the window and paper geometry.   |
| filename[#seq]                      | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: <b>current=la</b>                | The name of the surface converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. |

## DESCRIPTION

This program plots lightning strike data on a background map. The input to the program is a NLDN lightning file. Then, based on a plot domain and a variable selected by the user, a plot is generated. Overlay plots, contour plots and loops can also be generated if the appropriate options are set.

The programs starts off by prompting the user for input converted file name. The input file can either be in Albany or Unisys format. The default is Albany but Unisys can be specified with **in\_file** set to **unisys**. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **nldn** tag no matter whether **in\_file** is set to **unisys** or **albany**. The tag can be changed with the [in\\_file](#) resource. To generate a loop, the [num\\_hour](#) resource can be specified. This will tell the program to loop through this number of hours.

Next, the program prompts the user for the plot domain. This is done through a region menu which has predefined regions available. If needed a full plot domain can be specified to get a precise domain. The domain can be interactively changed using the mouse once the plot has been generated.

After the region has been set, the lightning data will be plotted. The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Initially, a map is plotted. Then, the lightning data file is opened and read in. A point is plotted for each strike. The strike data will be colored based on a variable type.

| Variable      | Description                   | Color Fill   |
|---------------|-------------------------------|--|
| <b>intens</b> | intensity of strike (default) | -500:blue, -300:lblue, -200:cyan, -175:lcyan, -150:green, -125:lgreen, -100:brown, -50:yellow, 0:red, 50:lred, 100:magenta, 150:lmagenta, 200:lgray, 300:white |
| <b>time</b>   | the minute of the strike      | 0:blue, 5:lblue, 10:cyan, 15:lcyan, 20:green, 25:lgreen, 30:brown, 35:yellow, 40:red, 45:lred, 50:magenta, 55:lmagenta   |
| <b>flash</b>  | the number of flashes         | 1:blue, 2:cyan, 3:green, 4:brown, 5:red, 6:magenta   |

**NOTE:** The Unisys format does not contain intensity or number of flash information.

## EXAMPLES

```
light -cu=la -re=mw -de=d
```

This plots a lightning strike map colored based in intensity.

**FILES**

**SEE ALSO**

# LOOPSET

## NAME

**loopset** - WXPloop setting program

## SYNOPSIS

**loopset** [*parameters*] [*command*]

## PARAMETERS

| Command Line         | Resource                   | Default             | Description   |
|----------------------|----------------------------|---------------------|---|
| <b>-h</b>            | <a href="#">help</a>       | No                  | Lists basic help information.   |
| <b>-df=filename</b>  | <a href="#">default</a>    | .wxpdef             | Sets the name of the resource file.   |
| <b>-na=name</b>      | <a href="#">name</a>       | heat                | Specifies the name used in resource file parsing.   |
| <b>-ba</b>           | <a href="#">batch</a>      | No                  | Run program in batch mode   |
| <b>-me=level</b>     | <a href="#">message</a>    | out2                | Specifies level of messages to be displayed   |
| <b>-wn=window</b>    | <a href="#">window_num</a> | 0                   | Specifies the window number. This sets up a unique route between this program and WXPloop so that WXPloop can be run multiple times simultaneously. |
| <b>-pa=parameter</b> | <a href="#">parameter</a>  | none                | Specifies additional parameters: <ul style="list-style-type: none"> <li>• <b>to=secs</b> - sets timeout on message queue</li> </ul>                 |
| <i>command</i>       | <a href="#">command</a>    | none<br>User prompt | Specifies the commands to be passed to the WXPloop program  |

## DESCRIPTION

The loopset program allows WXPloop's parameters to be changed. A IPC channel is used to communicate with the ingest program. When the **loopset** program establishes a connection with **wxploop**, the following prompt will appear:

```
Message queue opened (id:nnnnnn)
Connected to WXPloop program
Type help for help
Message:1:
```

At this point, you may enter a WXPloop command. To see the list of WXPloop commands, check the [wxploop](#) command documentation. The **loopset** program continues to accept dataset commands until "quit" or "exit" commands are entered. If a valid *command* is specified from the resource file or command line, only that command will be processed and the program will exit.

## Loopset Commands

There are some commands that are unique to **loopset**:

- **alloc** - returns the first available window number above 20. One that hasn't been allocated yet.
- **test** - tests to see if a window number is in use. Returns "found" or "not found".
- **clean** - deletes an existing window number. Use this if a WXPloop has terminated improperly.

## FILES

## SEE ALSO

# MAPPLT

## NAME

**mapplt** - Raw file and map plotting program

## SYNOPSIS

**mapplt** [*parameters...*] *filename*

## PARAMETERS

| Command Line             | Resource                    | Default  | Description  |
|--------------------------|-----------------------------|--|--|
| <b>-h</b>                | <a href="#">Help</a>        | No   | Lists basic help information.  |
| <b>-df=filename</b>      | <a href="#">default</a>     | .wxpdef  | Sets the name of the resource file.  |
| <b>-na=name</b>          | <a href="#">name</a>        | mapplt   | Specifies the name used in resource file parsing.  |
| <b>-ba</b>               | <a href="#">batch</a>       | No   | Run program in batch mode  |
| <b>-me=level</b>         | <a href="#">message</a>     | out2   | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>gridding information - <b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>      | <a href="#">file_path</a>   | current directory  | Specifies location of database files.  |
| <b>-rp=rawpath</b>       | <a href="#">raw_path</a>    | current directory  | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-ip=imagepath</b>     | <a href="#">image_path</a>  | current directory  | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>     | <a href="#">name_conv</a>   | name_conv  | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>       | <a href="#">in_file</a>     | raw  | Specifies the input file name tag. The default is <b>raw</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-cf=cityfile</b>      | <a href="#">city_file</a>   | sao.cty  | Specifies the name of the city database file used in finding station locations.  |
| <b>-mf=mapfile[,...]</b> | <a href="#">map_file</a>    | wxp.map or<br>cont.dat,<br>country.dat for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu=[hour la]</b>     | <a href="#">current</a>     | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=hour</b>          | <a href="#">hour</a>        | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-id=identifier</b>    | <a href="#">identifier</a>  | None   | Used to plot a specific observation. If not specified, all stations will be plotted.   |
| <b>-pd=domain</b>        | <a href="#">plot_domain</a> | User prompt<br>Batch: first<br>domain in<br>wxp.reg file.                      | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-re=region</b>        | <a href="#">region</a>      | Same as plot<br>domain   | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-pr=prior</b>         | <a href="#">stat_prior</a>  | Depends on<br>domain size  | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.   |

|  |                              |   |   |
|--|------------------------------|---|---|
| <b>-sc=scale_factor</b>                | <a href="#">plot_scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>                      | <a href="#">plot_format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.                      |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot_param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>              | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>                 | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels. |
| <b>-in=interval</b>                    | <a href="#">con_interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b>            | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.  |
| <b>-cb=base</b>                        | <a href="#">con_base</a>     | Lowest contour<br>value, maximum<br>wind speed                          | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.                                |
| <b>-fl=font_list</b>                   | <a href="#">font_list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>                 | <a href="#">color_table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>               | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the text color for labels above and below the plot.   |
| <b>-cod=color[:attr]</b>               | <a href="#">color_data</a>   | <b>green</b>  | Specifies the color for plotted data.   |
| <b>-com=color[:attr]</b>               | <a href="#">color_map</a>    | <b>red</b>  | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>              | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.  |
| <b>-cowl=color[:attr]</b>              | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for wind barbs.   |
| <b>-cocd=color[:attr]</b>              | <a href="#">color_cloud</a>  | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for cloud cover symbols.  |
| <b>-cowx=color[:attr]</b>              | <a href="#">color_wx</a>     | <b>yellow</b><br><b>brown</b> for<br>inverse plots                      | Specifies the color for weather symbols   |
| <b>-cofr=color[:attr],</b><br>...      | <a href="#">color_front</a>  | A predefined<br>color set   | Specifies the colors for fronts and pressure systems.   |
| <b>-cocm=color[:attr]</b>              | <a href="#">color_cmap</a>   | <b>black</b>  | Specifies the color for color fill contour maps.  |
| <b>-coco=color[:attr]</b>              | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for line contours.  |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color_fill</a>   | A predefined<br>color set based<br>on the <b>wxp.clr</b><br>color table | Specifies the color fill contour color set. See the <a href="#">color_fill</a> resource for more details.   |
| <b>-cola=color[:attr]</b>              | <a href="#">color_label</a>  | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the line contour label color.   |
| <b>-cocl=color[:attr]</b>              | <a href="#">color_clabel</a> | <b>black</b>  | Specifies the color fill contour label color.   |
| <b>-de=device[,...]</b>                | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                                    | Specifies the output device.  |
| <b>-ti=title</b>                       | <a href="#">title</a>        | Program name  | Specifies the window title.   |
| <b>-ic</b>                             | <a href="#">icon</a>         | <b>Off</b>  | Specifies whether to open window in iconified mode.   |
| <b>-bg=color</b>                       | <a href="#">background</a>   | <b>black</b>  | Specifies the background color of the window.   |

|   |                 |  |  |
|---|-----------------|--|--|
|   |                 | <b>white</b> for inverse plots                   |  |
| <b>-ge=geometry</b>   | <b>geometry</b> | <b>640x480</b>                                   | Specifies the window and paper geometry.   |
| {filename[-field] or<br>VH:label }<br>[filename[-field] or<br>VH:label ...] | <b>filename</b> | None<br>User Prompt<br>Batch: current= <b>la</b> | The first specification is for the name of the raw file to be plotted. More than one raw file/label can be listed.<br><br>The extra information following the "-" defines which field to use within a raw file. The field specification can be a column number or a field identifier. The field identifier is a string that accompanies a column header in the raw file or a string that precedes the data such as HGIRG=5.24<br><br>The second specification is a text label that will be positioned above or below the plot. This must be preceded with a two letter position identifier and a colon ( <i>VH:label</i> ) which is used to place the text on the plot. This identifier ( <i>VH</i> ) is one of the following: <b>ul,uc,ur,ll,lc,lr</b> which stands for upper left, upper center, upper right, lower left, lower center and lower right respectively. The text is also justified according to its location. |

## DESCRIPTION

This can be used for any of five purposes:

- Geographical Maps
- Annotating Plots
- Raw File Plotting
- City Database Plotting
- Station Identifier Plotting

Depending on the type, the program will prompt the user for the plot domain. This is done through a region menu which has predefined regions available. If needed a full plot domain can be specified to get a precise domain. The domain can be interactively changed using the mouse once the plot has been generated.

## GEOGRAPHICAL MAPS

Simple base maps can be drawn by specifying domain information via the **region** or **plot\_domain** resource. This is also handy for adding maps to overlay plot when the plot is complete rather than when the programs want to display the map.

## ANNOTATING PLOTS

Plots can be annotated by placing text labels above and below the plotting area. This is done by specifying labels on the command line.

*VH:label*

The location and alignment of the labels are specified a location prefix *VH:* followed by the text string. This will place text either above 'u' or below 'l' the plot, either left 'l', right 'r' or centered 'c'.

**NOTE:** When annotating, it is useful to put variable and plot type information upper left and date/time/forecast information upper right. Plot type information is up to the user. Date/time/forecast information can be generated by the **wxfile** program using **label** for the **output** resource.

## DISPLAYING RAW DATA

This program can be used to plot raw files that have been created with programs like **sfwx** and **upairwx**. If the user has created a raw data file, use this program to display the results. If the raw file contains multiple fields, all columns can be plotted or a specific one can be selected.

*filename[-field][:attributes]*

The extra information following the "-" defines which field to use within a raw file. The field specification can be a column number or a field identifier. The field identifier is a string that accompanies a column header in the raw file or a string that precedes the data such as "HGIRG=5.24". An additional set of plotting attributes can be appended to the filename. These are standard plotting attributes as described in the [Users Guide](#).

Also, a list of station identifiers can be specified using the **identifier** resource to limit plotting to only those sites. Also, the raw file may contain annotation and attributes information that can modify a station's output. If needed, the stations can be displayed with a marker such as a dot, plus sign or an asterisk instead of the identifier text. This can be done by specifying **mark:#** for the **plot\_param** resource.

## CITY DATABASE DISPLAY

Station locations can be plotted from a city database file. This can be handy in locating various reporting stations and for determining which stations have a certain plotting priority. If the *prior* is set to be greater than zero, those station identifiers with priorities less than or equal to *prior* will be displayed on the map as their three or four letter identifier. By default, these stations are plotted in the color specified by the **color\_data** resource. If necessary, the station color can be based on priority. If multiple colors are specified for the **color\_fill** resource, the first color will specify the color for priority 1 stations, the second color for priority 2 stations and so on. If a color is not specified for a priority, those stations will not plot. If needed, the stations can be displayed with a marker such as a dot, plus sign or an asterisk instead of the identifier text. This can be done by specifying **mark:#** for the **plot\_param** resource. The WMO number can be plotted by specifying **wmo**. The optional numeric column in the city database file can be plotted with the **num** parameter. This numeric field is often used for tabulating occurrence of reports by station. An example of plotting these data is:

```
mapplt -cf=citynum.dat -pr=7 -re=us -sc=.4 -pp=num
      -cof=white,yellow,green,cyan,magenta,blue,dgray
```

## SPECIFIC STATION PLOTTING

Plotting specific stations or specific locations can be done using the **identifier** resource. This can be used to plot the location of a specific set of stations either by searching for its location in the city databases or by specifying a latitude and longitude. Up to ten station identifiers can be specified. These identifiers will be looked up in the city database for locations, if location is not specified, and plotted. If the location is known, the latitude and longitude can be specified along with the station identifier as [*ident*]:*lat*:*lon*. If the station identifier is not specified, the ID defaults to a "+". With this capability, text can be placed on a plot. Also, annotation with attributes can be plotted by specifying:

[*ident*]:*lat*:*lon*] [*attributes*]:*data*]

The format for the *attributes* and *data* are the same as for the **plot\_param** resource. If needed, the stations can be displayed with a marker such as a dot, plus sign or an asterisk instead of the identifier text. This can be done by specifying **mark:#** for the **plot\_param** resource.

## EXAMPLES

```
mapplt -re=mw -de=d
```

This plots a map for the midwest region.

```
mapplt -dr=text ul:left_label lr:right_label
```

This labels a plot. If the labels contain spaces, they either have to be quoted or underscores can be used.

```
mapplt -re=31,-40,3 /data/hurricane/1997/ERIKA/track.raw-LINE
```

This plots the LINE field from the hurricane Erika track file. The region is selected to encompass the entire track.

```
mapplt -pd=cat,0,0,4,3,50,140 -pp=lalo:10:10,line \
/data/hurricane/1997/ERIKA/track.raw-WIND
```

This a category plot of wind speeds from hurricane Erika. A background grid every 10 points in each direction will be plotted. A line will be drawn connecting each point. The **plot\_domain** specification will plot data on a coordinate system that runs from 0,0 on the lower left to 50,140 in the upper right. The *clat* and *clon* parameters are the coordinates of the lower left corner of the plot. The *dx* and *dy* are the extent of the plot in each direction. This is different from standard map-type plot domains. Also, note that the X coordinate in this example is category and not X distance so that each item is plotted equally spaced in the order they appear in the raw file. The *nx* and *ny* (4,3) specifications are to set the aspect ratio of the plot as *nx-1* by *ny-1*.

```
mapplt -cf=sao.cty -pr=7 -re=us -sc=.4 \
-cof=white,yellow,green,cyan,magenta,blue,dgray
```

This will plot the stations of priority less than or equal to 7 on a US map. Each station is color coded based on its priority.

## FILES

- **sao.cty** - the surface station location database file

## SEE ALSO

- [grid](#) - the raw file gridding and contouring program
- [sfcwx](#) - the surface plotting program
- [upairwx](#) - the upper air plotting program

# MC2AREA

## NAME

**mc2area** - A filter for converting McIDAS packetized data to regular files

## SYNOPSIS

**mc2area** [-p *path*] *output\_filename*

## PARAMETERS

| Command Line           | Description  |
|------------------------|--|
| <b>-p=</b> <i>path</i> | Specifies the output path for data   |
| <i>output_filename</i> | Specifies the name of the output file. The path is prepended to the filename |

## DESCRIPTION

This program acts as a filter for data being ingested by the LDM on the McIDAS data feed. The McIDAS feed includes satellite imagery and other products such as profiler data in netCDF format which WXP can use. The McIDAS data is sent to the LDM in a packets and the FILE action on the LDM will not remove the packet information. Thus, there is the need for a filter to strip this information out.

Since this is a filter, the input is from standard input. The command line specifies only the output files. For example:

```
mc2area -p /data/profiler 97042200.prf < data/raw/97042200.prf_raw
```

The program can also be used directly by the LDM:

```
MCIDAS ^LWTOA3 14.
        PIPE -close /usr/local/bin/mc2area -p /data/ldm/mcidas %y%m%d%H.sve
MCIDAS ^LWFILE PROF
        PIPE -close /usr/local/bin/mc2area -p /data/ldm/profiler %y%m%d%H.prf
```

The output of the program is the unpackitized data. For satellite images, this is the compressed AREA file which can be used by **xsat**. For the profiler data, this is the netCDF data file which can be used by **prfplt**.

## FILES

- [AREA files](#)

## SEE ALSO

- [xsat](#)
- [prfplt](#)

# MOIST

## NAME

**moist** - Standard atmosphere computation program

## SYNOPSIS

**moist** [*parameters*] [*value*]

## PARAMETERS

| Command Line         | Resource                  | Default     | Description  |
|----------------------|---------------------------|-------------|--|
| <b>-h</b>            | <a href="#">help</a>      | No          | Lists basic help information.  |
| <b>-df=filename</b>  | <a href="#">default</a>   | .wxpdef     | Sets the name of the resource file.  |
| <b>-na=name</b>      | <a href="#">name</a>      | moist       | Specifies the name used in resource file parsing.  |
| <b>-ba</b>           | <a href="#">batch</a>     | No          | Run program in batch mode  |
| <b>-me=level</b>     | <a href="#">message</a>   | out2        | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• output values - <b>print</b></li> </ul> |
| <b>-pa=parameter</b> | <a href="#">parameter</a> | none        | Specifies additional parameters.   |
| <i>temp</i>          | <a href="#">value</a>     | none        | Specifies the input temperature in C   |
| <i>dewpoint</i>      |                           | User prompt | Specifies the input dewpoint in C  |
| <i>pressure</i>      |                           |             | Specifies the input surface pressure in mb   |
| <i>height</i>        |                           |             | Specifies the input station height in m.   |

## DESCRIPTION

The moisture variable calculation program derives various moisture variables given three of the following four variables: *temp,dewpoint,pressure,height*. If the station pressure is not known, it is determined from a standard atmosphere reduction using height. If any of the four variables *temp,dewpoint,pressure,height* are not specified on the command line, the user is prompted for the appropriate information. The resulting information displayed includes wetbulb temperature, vapor pressure, saturation vapor pressure, relative humidity, specific humidity, mixing ratio, saturation mixing ratio and equivalent potential temperature.

```

                                Results
Temperature =                    30.00 C
Dewpoint =                       20.00 C
Station pressure =                1000.00 mb
Wet bulb temperature =           22.87 C
Vapor pressure =                  23.36 mb
Saturation vapor pressure =       42.40 mb
Relative humidity =               55.08 percent
Specific humidity =               14.53 grams H2O /kg air
Mixing ratio =                    14.88 grams H2O /kg dry air
Saturation mixing ratio =         27.54 grams H2O /kg dry air
Equivalent potential temperature = 347.25 K

```

**NOTE:** Fahrenheit temperatures may be entered at the temperature and dewpoint prompts by adding 1000 to the temperature to be entered.

## FILES

## SEE ALSO

# PARSE

## NAME

**parse** - Text data parsing program

## SYNOPSIS

**parse** [*parameters...*] *filename*

## PARAMETERS

| Command Line                      | Resource                   | Default           | Description   |
|-----------------------------------|----------------------------|-------------------|---|
| <b>-h</b>                         | <a href="#">help</a>       | No                | Lists basic help information.   |
| <b>-df=<i>filename</i></b>        | <a href="#">default</a>    | <b>.wxpdef</b>    | Sets the name of the resource file.   |
| <b>-na=<i>name</i></b>            | <a href="#">name</a>       | <b>parse</b>      | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                        | <a href="#">batch</a>      | No                | Run program in batch mode   |
| <b>-me=<i>level</i></b>           | <a href="#">message</a>    | <b>out2</b>       | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>  |
| <b>-fp=<i>filepath</i></b>        | <a href="#">file_path</a>  | current directory | Specifies location of database files.   |
| <b>-dp=<i>datapath</i></b>        | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.  |
| <b>-nc=<i>name_conv</i></b>       | <a href="#">name_conv</a>  | <b>name_conv</b>  | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=<i>in_file</i></b>         | <a href="#">in_file</a>    | <b>raw_dat</b>    | Specifies the input file name tag. The default tag is <b>raw_dat</b> but will need to be modified for most applications. This can be determined from the product header if possible. Otherwise, it must be explicitly specified.  |
| <b>-cu=[<i>hour</i>]<b>la</b></b> | <a href="#">current</a>    | None              | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=<i>hour</i></b>            | <a href="#">hour</a>       | None              | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh=<i>num_hour</i></b>        | <a href="#">num_hour</a>   | <b>0</b>          | This specifies the number of hours that will be used. If this is not specified, a single hour will be parse. Otherwise a set of hours will be parsed.   |
| <b>-ph=<i>product</i></b>         | <a href="#">product</a>    | User Prompt       | This specifies the product header to search for.  |
| <b>-id=<i>identifier</i></b>      | <a href="#">identifier</a> | None              | Specifies the station to parse for. This is a string within a product that printing will start with. If this is not specified, the whole product will be displayed.   |
| <b>-pa=<i>param</i></b>           | <a href="#">parameter</a>  | None              | Specifies additional plotting parameters. See the <a href="#">parameter</a> resource for more details. Some possibilities are: <ul style="list-style-type: none"> <li><b>blank</b> -- stop parsing at a blank line</li> <li><b>3blank</b> -- stop parsing after 3 blank lines</li> <li><b>dollar</b> -- stop parsing at a dollar sign</li> <li><b>equal</b> -- stop parsing on a trailing equals sign</li> <li><b>line[=<i>lines</i>]</b> -- stop parsing after set number of lines (default 1)</li> <li><b>first</b> -- print only the first occurrence</li> <li><b>last</b> -- print only the last occurrence</li> <li><b>cont</b> -- keep file open to search for new products as they arrive</li> </ul> |

|                        |                 |  |  |
|------------------------|-----------------|--|--|
|                        |                 |  | <ul style="list-style-type: none"> <li>• <b>hdr</b> -- print just the headers</li> <li>• <b>prod</b> -- print just the product</li> <li>• <b>hdr+prod</b> -- print the header and the product</li> </ul> |
| <i>filename</i> [#seq] | <u>filename</u> | None<br>User Prompt<br>Batch: current= <b>la</b> | The name of the surface data file to be used. An optional sequence number can be added to designate the time for non-WXP files.  |

## DESCRIPTION

This program parses text data for a specific product and identifier. The input to the program is a raw ingested data file. The type of data file can be determined either from the product header or the in\_file resource. When a product is specified, it is cross-referenced against the **parse.lup** file to determine a file name tag to use. A sample of this lookup file is:

```
W      sev_dat
F      for_dat
C      cli_dat
. . .
```

If a product does not exactly match what is in the lookup file, a tag can be specified with the in\_file resource.

The programs starts off by prompting the user for input data file name. The user may specify the input file either via the command line or through the current resource. This will depend on the type of file either specified by the in\_file resource or the product header.

Next, the user enters a product header. The header can have wildcard characters to parse for multiple product types:

### Product Pattern Matching

|                                    |  |
|------------------------------------|--|
| . or ?                             | match a single character                       |
| - or *                             | match any character                            |
| [ <i>letters</i> ]                 | match a single character from the set.         |
| [ <i>^letters</i> ]                | match any character except those from the set. |
| ( <i>str1</i> [ <i> str2...</i> ]) | match strings                                  |
| _                                  | underscore matches a space.                    |
| / <i>secondline</i>                | second line parsing                            |

Second line parsing is also possible. For many products, the second line of the product is the AWIPS header:

```
** FPUS1 KIND 022030 ***
SFPIN
```

which in this case is "SFPIN". To parse for this, specify either "**FPUS1\_KIND**" or "**/SFPIN**".

If "**all**" is specified, all bulletins are searched.

Once the product header has been specified, the file will be opened and all products matching the given header will be displayed in their entirety.

## Selective Output

At times, the entire product is not desirable. By using a combination of the **identifier** resource and various output parameters, specific subsets of products can be displayed. By specifying a station identifier, the printing will start on a line that contains the identifier. Once an identifier is found, printing will continue until the end of product, unless otherwise specified. The identifier can be:

- *string* -- matches a string at the beginning of the line only

- `+string` -- matches if the string is contained anywhere within the line
- `%zone` -- matches if the zone matches a standard zone line (i.e. INZ029)
- `zn:zone` -- matches the zone
- `ua:id` -- matches a upper air ID

Printing normally continues to the end of product. To terminate it earlier, use one of the parameters in the **parameter** resource:

- **blank** -- stop parsing at a blank line
- **3blank** -- stop parsing after 3 blank lines
- **dollar** -- stop parsing at a dollar sign
- **equal** -- stop parsing on a trailing equals sign
- **line[=lines]** -- stop parsing after set number of lines (default 1)

Since more than one product can appear, it may be desirable to use only the first or last occurrence. Since products are continually appended to data files, it may be desirable to continue parsing even when the program has hit the end of file. This way the latest products will be printed as they are ingested. Additional parameters are available for these cases:

- **first** -- print only the first occurrence
- **last** -- print only the last occurrence
- **cont** -- keep file open to search for new products as they arrive

## Header Files

The use of a header file can considerably improve access to data files. Rather than parsing the entire file which at times is larger than 1MB, the product headers can be parsed directly out of a header file. Header files are much smaller and parse very fast. The header file contains a byte offset into the large file.

## EXAMPLES

To parse for the latest state forecast from KIND

```
parse -cu -nh=-12 -ph=FPUS1_KIND -pa=last
** FPUS1 KIND 022030 ***
SFPIN
INZ002>089-031000-

STATE FORECAST FOR INDIANA
NATIONAL WEATHER SERVICE INDIANAPOLIS IN
330 PM EST THU OCT 2 1997

.TONIGHT...FAIR AND WARMER. LOWS 50 TO 55.
.FRIDAY...MOSTLY SUNNY...BREEZY AND WARMER. HIGHS 80 TO 85.
.FRIDAY NIGHT...BECOMING MOSTLY CLOUDY. A CHANCE OF THUNDERSTORMS.
LOWS IN THE LOWER 60S.
.SATURDAY...MOSTLY CLOUDY...BREEZY AND A CHANCE OF THUNDERSTORMS.
WARM. HIGHS MIDDLE 70S TO AROUND 80.

.EXTENDED FORECAST...
.SUNDAY AND MONDAY...MOSTLY CLEAR AND WARM. LOWS MIDDLE 50S TO AROUND
60. HIGHS UPPER 70S TO LOWER 80S.
.TUESDAY...PARTLY CLOUDY AND MILD. LOWS AROUND 50 TO MIDDLE 50S.
HIGHS IN THE 70S.
DS
```

To parse for the latest state forecast using the AFOS PIL. Note the **in\_file** is specified since the product header does not appear in the **parse.lup** file.

```
parse -cu -nh=-12 -if=for_dat -ph=/SFPIN -pa=last
```

To parse for the latest zone forecast

```
parse -cu -nh=-12 -ph=FPUS53_KIND -id=%INZ029 -pa=dollar,last
** FPUS53 KIND 022040 COR ***
INZ020>023-028>030-035-036-043-044-051-052-060-067-030930-
CARROLL-CASS-CLAY-CLINTON-FOUNTAIN-KNOX-MIAMI-MONTGOMERY-PARKE-
SULLIVAN-TIPPECANOE-VERMILLION-VIGO-WARREN-WHITE-
INCLUDING THE CITIES OF...CRAWFORDSVILLE...FRANKFORT...LAFAYETTE...
LOGANSPORT...TERRE HAUTE...VINCENNES
330 PM EST THU OCT 2 1997

.TONIGHT...PARTLY CLOUDY AND WARMER. LOW IN THE MIDDLE 50S. SOUTHWEST
WIND 5 TO 10 MPH.
.FRIDAY...MOSTLY SUNNY AND WARMER. HIGH 80 TO 85. BREEZY SOUTHWEST
WIND 15 TO 20 MPH.
.FRIDAY NIGHT...BECOMING MOSTLY CLOUDY. A 40 PERCENT CHANCE OF
THUNDERSTORMS. MILD. LOW IN THE LOWER 60S.
.SATURDAY...MOSTLY CLOUDY...BREEZY AND A 40 PERCENT CHANCE OF
THUNDERSTORMS. MILD. HIGH IN THE UPPER 70S.
```

## FILES

- **parse.lup** - the parsing lookup file between headers and file name tags

## SEE ALSO

- [forecast](#) - the forecast parsing program

# PRFPLT

## NAME

**prfplt** - Profiler data plotting program

## SYNOPSIS

**prfplt** [*parameters...*] *filename*

## PARAMETERS

| Command Line             | Resource                    | Default  | Description  |
|--------------------------|-----------------------------|--|--|
| <b>-h</b>                | <a href="#">help</a>        | No   | Lists basic help information.  |
| <b>-df=filename</b>      | <a href="#">default</a>     | .wxpdef  | Sets the name of the resource file.  |
| <b>-na=name</b>          | <a href="#">name</a>        | prfplt   | Specifies the name used in resource file parsing.  |
| <b>-ba</b>               | <a href="#">batch</a>       | No   | Run program in batch mode  |
| <b>-me=level</b>         | <a href="#">message</a>     | out2   | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>   |
| <b>-fp=filepath</b>      | <a href="#">file_path</a>   | current directory  | Specifies location of database files.  |
| <b>-dp=datapath</b>      | <a href="#">data_path</a>   | current directory  | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>       | <a href="#">con_path</a>    | current directory  | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>       | <a href="#">raw_path</a>    | current directory  | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>      | <a href="#">grid_path</a>   | current directory  | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>     | <a href="#">image_path</a>  | current directory  | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>     | <a href="#">name_conv</a>   | name_conv  | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>       | <a href="#">in_file</a>     | prf_dat  | Specifies the input file name tag. The default is <b>prf_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-mf=mapfile[,...]</b> | <a href="#">map_file</a>    | wxp.map or<br>cont.dat,<br>country.dat for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu=[hour]la</b>      | <a href="#">current</a>     | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=hour</b>          | <a href="#">hour</a>        | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=num_hour</b>      | <a href="#">num_hour</a>    | 12   | This specifies the number of hours that will be used in the plot. If this is not specified, a 12 hour plot will appear.  |
| <b>-id=identifier</b>    | <a href="#">identifier</a>  | None<br>User prompt  | Specifies the station to plot. If specified, a time cross section will be plotted.   |
| <b>-pd=domain</b>        | <a href="#">plot_domain</a> | None<br>Batch: first   | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |

|  |                              |   |   |
|--|------------------------------|---|---|
|  |                              | domain in <b>wxp.reg</b> file.  |   |
| <b>-re=region</b>                      | <a href="#">region</a>       | Same as plot domain   | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-le=level</b>                       | <a href="#">level</a>        | None  | This specifies the level to use in horizontal cross-sections. By default, a time cross-section is plotted. If this resource is specified, a horizontal cross-section will be generated. |
| <b>-va=variable</b>                    | <a href="#">variable</a>     | User prompt<br>Batch: first variable listed in the <b>prfplt.var</b> file | This specifies the variable to plot along with its formatting attributes. A list of valid variables is given below.   |
| <b>-sc=scale_factor</b>                | <a href="#">plot_scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>                      | <a href="#">plot_format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.                              |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot_param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>              | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>                 | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.         |
| <b>-in=interval</b>                    | <a href="#">con_interval</a> | None for plots,<br>User prompt for grids. Batch: <b>0</b>                 | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.  |
| <b>-cb=base</b>                        | <a href="#">con_base</a>     | Lowest contour value, maximum wind speed                                  | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <b>con_base</b> resource.   |
| <b>-fl=font_list</b>                   | <a href="#">font_list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>                 | <a href="#">color_table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>               | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse plots.                           | Specifies the text color for labels above and below the plot.   |
| <b>-cod=color[:attr]</b>               | <a href="#">color_data</a>   | <b>green</b>  | Specifies the color for plotted data.   |
| <b>-com=color[:attr]</b>               | <a href="#">color_map</a>    | <b>red</b>  | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>              | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.  |
| <b>-cowl=color[:attr]</b>              | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse plots.                           | Specifies the color for wind barbs.   |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color_fill</a>   | A predefined color set based on the <b>wxp.clr</b> color table            | Specifies the color fill contour color set. See the <b>color_fill</b> resource for more details.  |
| <b>-de=device[,...]</b>                | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                                      | Specifies the output device.  |
| <b>-ti=title</b>                       | <a href="#">title</a>        | Program name  | Specifies the window title.   |
| <b>-ic</b>                             | <a href="#">icon</a>         | <b>Off</b>  | Specifies whether to open window in iconified mode.   |
| <b>-bg=color</b>                       | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots                            | Specifies the background color of the window.   |
| <b>-ge=geometry</b>                    | <a href="#">geometry</a>     | <b>640x480</b>  | Specifies the window and paper geometry.  |
| <b>filename[#seq]</b>                  | <a href="#">filename</a>     | None  | The name of the surface converted data file to be plotted. An   |

|  |  |  |
|--|--|--|
|  | User Prompt<br>Batch: current= <b>la</b> | optional sequence number can be added to designate the time for non-WXP files. |
|--|--|--|

## DESCRIPTION

This program plots profiler data. There are two types of output for the program. The first is a time cross-section using a series of profiler data files. The second mode is a horizontal cross-section at a specified level. By default, a time cross-section will be plotted. If a level is specified with the [level](#) resource, a horizontal cross-section will be plotted. surface data on a background map. The input to the program is a profiler data file.

The programs starts off by prompting the user for input profiler file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **prf\_dat** tag but this can be changed with the [in\\_file](#) resource. For time cross-sections, the [num\\_hour](#) resource is used to set the number of hours in the cross-section. The default is 12 hours.

For time cross-sections, the program will now prompt the user for a profiler station to use.

The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the **prfplt.var** file. The internal variables are:

### Internal Variables

| Variable                                   | Description and units   |
|--|---|
| <b>city</b>                                | All stations in the data file   |
| <b>temp</b>                                | Temperature (C) -- surface level only   |
| <b>dewp</b>                                | Dewpoint (C) -- surface level only  |
| <b>dewd</b>                                | Dewpoint depression (C) -- surface level only   |
| <b>wdir</b>                                | Wind direction (deg)  |
| <b>wspd</b>                                | Wind speed (knt)  |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)  |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)  |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)  |
| <b>vvel</b>                                | Vertical velocity (m/s)   |
| <b>shear</b>                               | Vertical wind shear (/s) -- layer can be adjusted with level resource                             |
| <b>hel</b>                                 | 0 - 3000 mb AG storm relative helicity ( $m^2/s^2$ ) -- layer can be adjusted with level resource |

These parameters can be accessed by name if they have not been redefined in the **prfplt.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the profiler file is opened and data read in. First, the background map will plot. Any additional computations are performed and the data are plotted. If a simple plot is specified such as "**temp**", the data will be plotted to 0 decimal places on the map. Formatting of the number can be done with the **plot\_format** resource. Composite plots can either be specified with the **variable** resource (.var file, resource file, command line or variable prompt). The **variable** resource allows the user to control much of the data formatting.

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## EXAMPLES

```
prfplt -cu=la -id=PLTC2 -va=wbrb -de=d
```

This generates a 12 hour time cross-section of wind data plotted as wind barbs for the profiler site PLTC2.

```
prfplt -cu=1a -le=3000m -va=wbrb -de=d
```

This will generate a horizontal cross-section of wind data at the 3000m level. All profiler stations are plotted.

## **FILES**

- **prfplt.var** - variable menu and definitions

## **SEE ALSO**

# PSCAT

## NAME

**pscat** - Postscript concatenation program

## SYNOPSIS

**pscat** [*options...*] [*file...*]

## PARAMETERS

| Options                    | Description   |
|----------------------------|---|
| <b>-h</b>                  | Lists help information.   |
| <b>-port</b>               | Specifies to print plot in portrait mode.   |
| <b>-eps</b>                | Specifies to create an Encapsulated Postscript file (EPS).  |
| <b>-title</b> <i>title</i> | Specifies title line in EPS file. This is useful for identifying graphics plots with word processors that list title in EPS graphics frame. |
| <b>-output</b> <i>file</i> | Specifies output print file or device.  |
| <i>file</i>                | Input Postscript stub   |

## DESCRIPTION

This program adds Postscript header and trailer information onto files to prepare them for printing. This is used when the Postscript append option has been specified with the **device** resource. The Postscript header contains the Postscript definition line plus any macro definitions. Once the graphic has been completely entered into the file, a trailer "**showpage**" command must be added in order to end processing for that page and print it out. Obviously, only one header and trailer can be contained in a particular Postscript file for a WXP image. By default when WXP creates a Postscript plot, this header is included:

```

%!PS-Adobe-1.0
%%Title: WXP Graphics Display
%%Creator: WXP Postscript Processor-2.0
%%CreationDate: Mon Sep 29 02:07:55 1997
%%EndComments
%%Postscript initialization
30 dict begin
/saveobj save def
/inch {72 mul} def
/dot {.33 mul} def
/mv {moveto} def
/ln {lineto} def
/lw {setlinewidth} def
/np {newpath} def
/cp {closepath} def
90 rotate
.25 inch -8.25 inch translate
1 dot lw
0 setgray
2 setlinejoin
[] 0 setdash
%%Begin plot
... ( insert stub here)
saveobj restore
showpage
end

```

```
%%EOF
```

If WXP plots are to be overlaid, this information cannot be saved. The append option for WXP will accomplish this. Plot information is therefore appended until the desired information is contained in the file. In order to prepare the appended file for printing **pscat** is run which puts the above header and trailer on the file. Now the file can be printed properly.

If no output filename is given, **pscat** reads from standard input. Otherwise, **pscat** will read each file from the command line, printing the header before the first file and placing the trailer after the last file. The output of **pscat** is standard output unless the **output** parameter is specified or the **wxpps\_print** environment variable is set. All options for output (file,pipe) can be set by specifying:

```
setenv wxpps_print postscript_file  
setenv wxpps_print '|program'
```

The output parameter will also recognize the output types along with "-" for output to standard output.

## SEE ALSO

- [hpglcat](#) - HPGL concatenation program
- [gifcat](#) - GIF file concatenation program

# RAD

## NAME

**rad** - Radar display program

## SYNOPSIS

**rad** [*parameters...*] *filename*

## PARAMETERS

| Command Line         | Resource                   | Default           | Description   |
|----------------------|----------------------------|-------------------|---|
| <b>-h</b>            | <a href="#">help</a>       | No                | Lists basic help information.   |
| <b>-df=filename</b>  | <a href="#">default</a>    | .wxddef           | Sets the name of the resource file.   |
| <b>-na=name</b>      | <a href="#">name</a>       | rad               | Specifies the name used in resource file parsing.   |
| <b>-ba</b>           | <a href="#">batch</a>      | No                | Run program in batch mode   |
| <b>-me=level</b>     | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• watch box details - <b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>  | <a href="#">file_path</a>  | current directory | Specifies location of database files.   |
| <b>-dp=datapath</b>  | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files which includes <b>front</b> files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>   | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.  |
| <b>-rp=rawpath</b>   | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.   |
| <b>-gp=gridpath</b>  | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.   |
| <b>-ip=imagepath</b> | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.   |
| <b>-nc=name_conv</b> | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-inp=input</b>    | <a href="#">input</a>      | mdr               | Specifies the input file type. The default is <b>mdr</b> . Possible values are: <ul style="list-style-type: none"> <li>• <b>mdr</b> - MDR or RCM radar data from converted file (Default). Name convention set to <b>rad_cvt</b>.</li> <li>• <b>raw</b> - Raw MDR data from ingested radar file Name convention set to <b>rad_dat</b>.</li> <li>• <b>rcm</b> - RCM summary data. Name convention set to <b>rcm_cvt</b>.</li> <li>• <b>rcmsite</b> - RCM single site data. Name convention set to <b>rcm_dat</b>.</li> <li>• <b>nids</b> - NIDS data input and Unisys radar composite Name convention set to <b>nids</b>.</li> <li>• <b>nowrad</b> - NOWRad composite plot Name convention set to <b>nowrad</b>.</li> <li>• <b>rrwds</b> - Old RRWDS files Name convention set to <b>rrwds</b>.</li> </ul> |
| <b>-if=in_file</b>   | <a href="#">in_file</a>    | rad_cvt           | Specifies the input file name tag. The default is <b>rad_cvt</b> , but it can be modified to any value in the filename convention file.   |

|   |                             |  |  |
|---|-----------------------------|--|--|
|   |                             |  | <p>A full name convention can be specified as well.<br/>The input file type can be changed as well:</p> <ul style="list-style-type: none"> <li>• <b>mdr</b> - MDR radar data (Default)</li> <li>• <b>rcm</b> - RCM summary data</li> <li>• <b>rcmsite</b> - RCM single site radar data</li> <li>• <b>nids</b> - NIDS data input and Unisys radar composite</li> <li>• <b>nowrad</b> - NOWRad composite plot</li> <li>• <b>rrwds</b> - Old RRWDS files</li> </ul> |
| <b>-of</b> = <i>out_file</i> [, <i>out...</i> ]   | <a href="#">out_file</a>    | None   | Reserved for future use.   |
| <b>-cf</b> = <i>cityfile</i>                      | <a href="#">city_file</a>   | <b>rad.cty</b>   | Specifies the name of the city database file used in finding station locations.  |
| <b>-mf</b> = <i>mapfile</i> [,...]                | <a href="#">map_file</a>    | <b>wxp.map</b> or <b>cont.dat</b> , <b>country.dat</b> for domains outside North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu</b> =[ <i>hour</i>   <b>la</b> ]           | <a href="#">current</a>     | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho</b> = <i>hour</i>                          | <a href="#">hour</a>        | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh</b> = <i>num_hour</i>                      | <a href="#">num_hour</a>    | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.  |
| <b>-id</b> = <i>identifier</i>                    | <a href="#">identifier</a>  | User prompt  | Used to plot a specific RCM site. If not specified, user will be prompted for the ID.  |
| <b>-pd</b> = <i>domain</i>                        | <a href="#">plot_domain</a> | User prompt<br>Batch: first domain in <b>wxp.reg</b> file.                               | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-re</b> = <i>region</i>                        | <a href="#">region</a>      | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-va</b> = <i>variable</i>                      | <a href="#">variable</a>    | User prompt<br>Batch: first variable listed in the <b>rad.var</b> file                   | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.   |
| <b>-sc</b> = <i>scale_factor</i>                  | <a href="#">plot_scale</a>  | <b>1.0</b>   | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm</b> = <i>format</i>                        | <a href="#">plot_format</a> | None   | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp</b> = <i>param</i><br>[, <i>param...</i> ] | <a href="#">plot_param</a>  | None   | <p>Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.</p> <p><b>MDR/RCM</b></p> <ul style="list-style-type: none"> <li>• <b>site</b> -- plots location of radar site</li> <li>• <b>dist=#</b> -- plots a single distance ring</li> <li>• <b>nodist</b> -- disables distance rings</li> <li>• <b>nostatus</b> -- disables plotting of radar status (NE,NA,OM)</li> </ul> <p><b>NOWRad</b></p>                |

|   |                              |  |  |
|---|------------------------------|--|--|
|   |                              |  | <ul style="list-style-type: none"> <li><b>run=#</b> -- specifies run length for plotting areas (def=1, 2 for lat-lon domains)</li> </ul> <p><b>NIDS</b></p> <ul style="list-style-type: none"> <li><b>site</b> -- plots location of radar site</li> <li><b>dist=dist1:dist2:dist3</b> -- plots distance rings at specified distances</li> <li><b>nodist</b> -- disables distance rings</li> <li><b>nograph</b> -- disables plotting CREF storm chart</li> <li><b>graph</b> -- enables plotting CREF storm chart (default)</li> <li><b>clutter=rad</b> -- does not plot echoes within radius to eliminate ground clutter</li> <li><b>noclutter</b> -- disables clutter elimination</li> </ul> |
| <b>-dr=drawable</b> [,...]                            | <a href="#">draw</a>         | <b>all</b>   | Specifies which part of the plot is to be drawn.   |
| <b>-la=label</b> [,...]                               | <a href="#">label</a>        | Program defaults   | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-in=interval</b>                                   | <a href="#">con interval</a> | None for plots, User prompt for grids. Batch: <b>0</b>         | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.   |
| <b>-cb=base</b>                                       | <a href="#">con base</a>     | Lowest contour value, maximum wind speed                       | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <b>con_base</b> resource.  |
| <b>-fl=font_list</b>                                  | <a href="#">font list</a>    | <b>modern.fnt</b>  | Specifies a list of fonts to use in the plots.   |
| <b>-ct=color_table</b>                                | <a href="#">color table</a>  | <b>wxp.clr</b>   | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |
| <b>-cot=color</b> [:attr]                             | <a href="#">color text</a>   | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the text color for labels above and below the plot.  |
| <b>-cod=color</b> [:attr]                             | <a href="#">color data</a>   | <b>green</b>   | Specifies the color for plotted data.  |
| <b>-com=color</b> [:attr]                             | <a href="#">color map</a>    | <b>red</b>   | Specifies the color for background maps.   |
| <b>-coln=color</b> [:attr]                            | <a href="#">color line</a>   | <b>magenta:st=dsh</b>  | Specifies the color for lat/lon lines.   |
| <b>-cowd=color</b> [:attr]                            | <a href="#">color wind</a>   | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the color for wind barbs.  |
| <b>-cowa=color</b> [:attr]                            | <a href="#">color watch</a>  | <b>white</b><br><b>black</b> for inverse plots                 | Specifies the color for watch boxes.   |
| <b>-cof=[val:]color</b> [:attr],...                   | <a href="#">color fill</a>   | A predefined color set based on the <b>wxp.clr</b> color table | Specifies the color set for use in plotting the data. See the <b>color_fill</b> resource for more details. Default is <b>none, blue, cyan, green, yellow, red, magenta</b> for MDR/RCM plots.  |
| <b>-de=device</b> [,...]                              | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                           | Specifies the output device.   |
| <b>-ti=title</b>                                      | <a href="#">title</a>        | Program name   | Specifies the window title.  |
| <b>-ic</b>  | <a href="#">icon</a>         | <b>Off</b>   | Specifies whether to open window in iconified mode.  |
| <b>-bg=color</b>                                      | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots                 | Specifies the background color of the window.  |
| <b>-ge=geometry</b>                                   | <a href="#">geometry</a>     | <b>640x480</b>   | Specifies the window and paper geometry.   |
| <b>filename</b> [#seq]<br><b>[filename</b> [#seq]...] | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: <b>current=la</b>                | The name of the radar data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified to generate a loop.   |

## DESCRIPTION

This program plots various types of radar data. Each type will be discussed below:

### MDR (Manually Digitized Radar)

MDR data is broadcast once an hour (around 30 after) and is a low resolution radar composite for the contiguous US. The input to **rad** is a converted data file produced by the [radcvt](#) program. Then, based on a plot domain and a variable selected by the user, a plot is generated. Overlay plots, contour plots and loops can also be generated if the appropriate options are set.

The program starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **rad\_cvt** tag but this can be changed with the [in\\_file](#) resource. To generate a loop, the [num\\_hour](#) resource can be specified. This will tell the program to loop through this number of hours.

Next, the program prompts the user for the plot domain. This is done through a region menu which has predefined regions available. If needed a full plot domain can be specified to get a precise domain. The domain can be interactively changed using the mouse once the plot has been generated.

After the region has been set, the city database is read in and a cross-reference table is setup. Since converted data is not saved with location information, the program will refer to this table to location each station in the input file.

The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the **rad.var** file. The internal variables are:

#### Internal Variables

| Variable        | Description and units  | Plots   |
|-----------------|--|---|
| <b>city</b>     | All stations in the data file  | text  |
| <b>stat</b>     | Radar status: <b>NA, NE, OM</b>  | text  |
| <b>pr_typ</b>   | Precipitation type and intensity (e.g. <b>RW-</b> )<br>Intensity is based on max radar intensity:<br>1=-, 2=, 3=+, 4=++, 5=x, 6=xx | text  |
| <b>pr_tnd</b>   | Precipitation trend: +=increase, -=decrease, <b>NC</b> =no change, <b>NEW</b> =new   | text  |
| <b>pr_tt</b>    | Precipitation type and trend (type and single char for trend):<br>/=increase, \=decrease, -=no change, *=new                       | text  |
| <b>echtop</b>   | Echo tops (100 ft). Top positioned at location. If top has a star "*", no location is available and top positioned over station.   | text  |
| <b>pr_amv</b>   | Area movement, direction and speed (knt)   | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>pr_cmvmv</b> | Cell movement, direction and speed (knt)   | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>watch</b>    | Watch boxes (severe and tornado)   |   |

These parameters can be accessed by name if they have not been redefined in the **rad.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

There are several derived variables from this list which are defined in the **rad.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable   | Description  | Plots |
|------------|--|-------|
| <b>all</b> | All data, a composite plot of status, type and trend, area movement (vector), cell movement (barb), echo tops, and watches | comp  |

|            |  |      |
|------------|--|------|
| <b>sum</b> | Only radar status. Useful for just plotting radar summary. | comp |
|------------|--|------|

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the converted file is opened and data read in and radar summary plotted. The summary is plotted with the following color scheme. The scheme can be changed using the [color\\_fill](#) resource.

| Intensity | DBz   | Color         | Precip Intensity   | Approx. Rainfall Rate (inches/hour) |            |
|-----------|-------|---------------|--------------------|-------------------------------------|------------|
|           |       |               |                    | Stratiform                          | Convective |
| <b>1</b>  | 20-30 | Blue/DGreen   | "-" -- Light       | 0.0 to 0.1                          | 0.0 to 0.2 |
| <b>2</b>  | 30-40 | Cyan/Green    | " " -- Moderate    | 0.1 to 0.5                          | 0.2 to 1.1 |
| <b>3</b>  | 40-45 | Green/LGreen  | "+" -- Heavy       | 0.5 to 1.0                          | 1.1 to 2.2 |
| <b>4</b>  | 45-50 | Yellow/Yellow | "++" -- Very Heavy |                                     | 2.2 to 4.5 |
| <b>5</b>  | 50-55 | Red/Orange    | "x" -- Intense     |                                     | 4.5 to 7.1 |
| <b>6</b>  | >55   | Magenta/Red   | "xx" -- Extreme    |                                     | > 7.1      |

The background map will plot and then any overlay data. Composite plots can either be specified with the variable resource (.var file, resource file, command line or variable prompt). The variable resource allows the user to control much of the data formatting.

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

**INTERPRETIVE NOTE:** When determining the intensity of precipitation within a MDR radar grid box, the maximum intensity within the box is the intensity reported. Therefore, precipitation coverage is often overestimated, especially in cases of convective precipitation. Also, since most stratiform precipitation occurs below 10,000 feet, radars may not fully report large areas of stratiform rain and especially snow.

### Watch Box Data

The watch box is determined based on the coordinates of two stations. These coordinates or station locations are looked up in the **sao\_all.cty** file. This file contains some common station coordinates that are used in watch box location but may not report surface data. At times, the stations are not in the database file and the watch box will not plot. You can add stations to the **sao\_all.cty** file if needed.

Second, the watch box is associated with a valid time range. The box will only be plotted if the radar summary is within the valid time. This is NOT determined by the current time.

The box is plotted along with two lines of text. For example:

```
S716
to 3Z
```

The first line gives the watch type: **S** for severe thunderstorm or **T** for tornado. The number represents the watch number which can be cross referenced to the watch data. Watch data is printed as such:

```
Date: 2235Z 27 AUG 97
WATCH_decode_file: file not found: /rainbow/data/ddplus/12082797.wws
Reading watch file: /rainbow/data/ddplus/18082797.wws
Watch: 716 SEVERE : VALID
Watch: 716 SEVERE : VALID
If the message level is out3, the following information is printed:
Reading watch file: /rainbow/data/ddplus/18082797.wws
Watch: 716 SEVERE : Beg: 97/08/27 21:00 End: 97/08/28 03:00 VALID
Width:60 NORTH Location: 35W GGW to 95E ISN
```

Aviation location: 30W GGW to 12WSW MOT  
 Watch: 716 SEVERE : Beg: 97/08/27 21:00 End: 97/08/28 03:00 VALID  
 Width:60 NORTH Location: 35W GGW to 95E ISN  
 Aviation location: 30W GGW to 12WSW MOT

This will display beginning and ending times as well as location information so the user can plot the data by hand. The width is in nautical miles and the direction is either **EAST** for east-west of a line, **NORTH** for north-south of a line, or **EITHER** for either side of a line.

### RCM Radar Summary Display

RCM (Radar Coded Messages) are similar to the MDR reports. RCM data is on a grid 1/4<sup>th</sup> the size of the MDR grid or roughly 12km resolution. RCM summaries are generated from the individual site RCMs using the [radcv](#) program and placed in a WXP radar converted file. The program determines the type of converted file from the contents. If the converted data is MDR data, MDR plotting procedures will be used. If the converted file is RCM data, then RCM plotting will occur. The [radcv](#) program will save the RCM data with the file tag **rcm\_cvt**. Using **rcm** for the **input** resource will make this the default tag. The procedure for running the program is nearly identical to the MDR procedure. RCM summaries use the same echo range (1 to 6) as does MDR. The radar summary plot may be annotated with various pieces of information such as storm ID, echo tops, movement, hail, TVS (Tornado Vortex Signatures), mesocyclone location, and severe weather watch boxes:

#### Internal Variables

| Variable      | Description and units           | Plots   |
|---------------|---------------------------------|---|
| <b>city</b>   | Site name                       | text  |
| <b>stat</b>   | Site status                     | text=NA,NE,OM                                       |
| <b>mode</b>   | Site mode (Precip,Clear Air)    | text=P,C  |
| <b>maxtop</b> | Site maximum tops (100ft)       | value   |
| <b>tv</b>     | Tornado Vortex Signature (TVS)  | value=1, text=TVS                                   |
| <b>meso</b>   | Mesocyclone                     | value=1,text=MESO                                   |
| <b>id</b>     | Storm ID                        | text  |
| <b>top</b>    | Echo tops in 100s ft            | value   |
| <b>spd</b>    | Speed of movement in knots      | value   |
| <b>hail</b>   | Hail (0=none,1=prob, 2=likely)  | value=hail,<br>hail=2, text=*,<br>hail=1, text=+    |
| <b>mvmt</b>   | Direction and speed of movement | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |

### RCM Site Radar

This program will display RCM single site data. When the **input** resource is set to **rcmsite**, the **rad** program will read the raw RCM data and plot the RCM data for a specific site. The program will prompt the user for the site ID, if it is not specified in the **identifier** resource. The domain is based on the radar site. If the user specifies a domain, the data will be remapped to the new domain. The site data can be annotated with storm attribute information if the **variable** resource is set to **comp**. The data plotted are:

| Type          | Description                    |
|---------------|--------------------------------|
| Red triangle  | Tornado Vortex Signature (TVS) |
| Yellow circle | Mesocyclone                    |
| Green *       | Hail likely                    |
| Green x       | Hail possible                  |
| Wind barb     | Cell movement                  |
| Text UL       | Storm ID                       |
| Text UR       | Echo tops                      |

### Radar Mosaics (NOWRad and Unisys)

Radar Mosaics are high resolution (2, 4, and 8 km) national composites of all Nexrad sites. NOWRad data is developed by WSI and comes in two forms: 2km resolution data on a latitude longitude grid (MASTER sector) and 8km resolution on a lambert conformal grid (US sector) both based on base reflectivity. The **input** resource is set to **nowrad** to tell **rad** that the input data is NOWRad. These are put into files defined by the **nowrad** and **nowrad\_us** file name tags.

The Unisys mosaics are on a lambert conformal grid, in various resolutions and include base reflectivity, echo tops and precipitation. These are in the same format as the composite reflectivity NIDS data so the **nids** file tag is used. The program will prompt the user for the input file name. This can be entered via the command line or using the **current** resource. To generate a loop, the **num\_hour** resource can be specified. This will tell the program to loop through this number of hours.

Once the file is determined, the user will be prompted for the device and plotting will begin. The color scheme is listed below. The last two schemes require the **radar.clr** color table.

| dbZ range | Default Color | nowrad.cfl | nowrad_std.cfl |
|-----------|---------------|------------|----------------|
| <5        | off           | off        | off            |
| 5-10      | blue          | DDGreen    | XGreen         |
| 10-15     | lblue         | DGreen     | BXGreen        |
| 15-20     | cyan          | DMGreen    | BGreen         |
| 20-25     | lcyan         | MGreen     | MBGreen        |
| 25-30     | green         | MBGreen    | MGreen         |
| 30-35     | lgreen        | BGreen     | DMGreen        |
| 35-40     | brown         | BXGreen    | DGreen         |
| 40-45     | yellow        | XGreen     | DDGreen        |
| 45-50     | red           | Yellow     | Yellow         |
| 50-55     | lred          | Orange     | Orange         |
| 55-60     | magenta       | XRed       | XRed           |
| 60-65     | lmagenta      | BRed       | BRed           |
| 65-70     | dgray         | MRed       | MRed           |
| 70-75     | lgray         |            |                |
| 75-80     | white         |            |                |

**NOTE:** The 2 km sectors sector plots are high resolution and often take a long time to plot. Smaller regions will plot faster since echoes outside the domain do not plot. For large regional plots, it is recommended to use the lower resolution (4, 8 km) mosaics.

### NIDS Data

The NIDS data feed contains many types of images from each WSR-88D radar. The images are for a single radar site and include the following:

| Name  | Description                     | Type              |
|-------|---------------------------------|-------------------|
| bref1 | Base reflectivity, level 1      | radial            |
| bref2 | Base reflectivity, level 2      | radial            |
| bref3 | Base reflectivity, level 3      | radial            |
| bref4 | Base reflectivity, level 4      | radial            |
| lref1 | Layer reflectivity, level 1     | rastor            |
| lref2 | Layer reflectivity, level 2     | rastor            |
| lref3 | Layer reflectivity, level 3     | rastor            |
| cref  | Composite reflectivity, level 1 | rastor, with text |
| vel1  | Base velocity, level 1          | radial            |
| vel2  | Base velocity, level 2          | radial            |

|       |                                       |                                |
|-------|---------------------------------------|--------------------------------|
| vel3  | Base velocity, level 3                | radial                         |
| vel4  | Base velocity, level 4                | radial                         |
| srmv1 | Storm relative mean velocity, level 1 | radial                         |
| srmv2 | Storm relative mean velocity, level 2 | radial                         |
| tops  | Echo tops                             | rastor                         |
| vil   | Vertical integrated liquid            | rastor                         |
| pre1  | 1 hour precipitation total            | rastor                         |
| pret  | Storm total precipitation             | rastor                         |
| prea  | Digital precipitation array           | rastor, 1/40th LFM grid        |
| vad   | Vertical azimuth display              | winds aloft time cross section |

To access these files, it is recommended to use the above names as file name extensions such as **9707241255.bref1**. Then allot the name convention to use a extension (%e) and use **nids\_bref1** for the input file type. The **input** resource must be set to **nids** in order for **rad** to know that the data are in NIDS format.

After the file type is determined, the program will prompt for the input file name. This can be entered via the command line or using the **current** resource. To generate a loop, the **num\_hour** resource can be specified. This will tell the program to loop through this number of hours.

The program will then display the image and overlay a map. If no plot domain is specified, the image is plotted in a polar stereographic projection sized so the image fits the window. A plot domain can be specified to remap the image for overlay. The remap will only work with polar stereographic and lat/lon projections.

The default image colors are dependent on the image type:

- **bref, cref, lref, tops, vil, pre** - uses a standard set of colors but the cutoffs are based on the threshold values in the NIDS file. The colors are "**off, blue, lblue, cyan, lcyan, green, lgreen, brown, yellow, red, lred, magenta, lmagenta, dgray, lgray, white**".
- **vad** - the wind barbs are color coded base on RMS of the variability of winds over the reporting period. These values range from 0-16 in intervals of 4. The colors are "**lblue, lcyan, lgreen, yellow, lred**".
- **vel** - uses a preset color scheme for consistency between plots. The color scheme is: "**-70:blue, -63:lblue, -49:cyan, -35:lcyan, -25:green, -19:lgreen, -9:dgray, 0:off, 1:lgray, 10:yellow, 20:brown, 26:lred, 36:red, 50:lmagenta, 64:magenta**".
- **srmv** - uses a preset color scheme for consistency between plots. The color scheme is: "**-50:lblue, -40:cyan, -30:lcyan, -22:green, -10:lgreen, -5:dgray, -1:off, 0:lgray, 5:yellow, 10:brown, 22:lred, 30:red, 40:lmagenta, 50:magenta**".

This can be changed using the **color\_fill** resource.

### Storm Attributes Plotting

The composite reflectivity product contains the storm attributes table. By default, only the reflectivity plot is displayed. If the **variable** resource is specified, attributes will be plotted over the reflectivity. The possible variables are:

#### Internal Variables

| Variable    | Description and units          | Plots  |
|-------------|--------------------------------|--|
| <b>id</b>   | Storm ID                       | text   |
| <b>tv</b>   | Tornado Vortex Signature (TVS) | value=1,<br>text=TVS                               |
| <b>meso</b> | Mesocyclone                    | value=1,text=M<br>ESO                              |
| <b>posh</b> | Probability of severe hail.    | value=prob,<br>prob>75, text=*,<br>prob>50, text=X |

|              |                                 |                                |   |
|--------------|---------------------------------|--------------------------------|---|
| <b>poh</b>   | Probability of hail.            |                                | same as posh  |
| <b>hsize</b> | Hail size in inches             |                                | value   |
| <b>dbzm</b>  | Maximum Dbz value               |                                | value   |
| <b>dbzht</b> | Height of max. Dbz              |                                | value   |
| <b>top</b>   | Echo tops in 100s ft            |                                | value   |
| <b>dir</b>   | Direction of movement           |                                | value   |
| <b>spd</b>   | Speed of movement in knots      |                                | value   |
| <b>mvmt</b>  | Direction and speed of movement |                                | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>comp</b>  | <b>Type</b>                     | <b>Description</b>             |   |
|              | Red triangle                    | Tornado Vortex Signature (TVS) |   |
|              | Yellow circle                   | Mesocyclone                    |   |
|              | Green *                         | > 75% Hail                     |   |
|              | Green x                         | > 50% Hail                     |   |
|              | Wind barb                       | Cell movement                  |   |
|              | Text UL                         | Storm ID                       |   |
|              | Text UR                         | Echo tops                      |   |

These parameters can be accessed by name if they have not been redefined in the **nids.var** file.

### RRWDS Data

Remote Radar Weather Display System (RRWDS) radar site displays can be handled by specifying **rrwds** for the **input** resource. The radar can be remapped for overlay purposes.

### EXAMPLES

```
rad -cu=la -re=mw -va=all -de=d
```

This plots a radar summary overlaid with a composite of MDR data including watch boxes.

```
rad -if=nowrad -cu=la -re=40,-90,.4 -mf=fi:map_cnty -de=d
```

This will plot the latest NOWRad MASTER sector for a location in Illinois and overlay it with a county map.

```
rad -if=nids_bref1 -cu=la -de=d
```

This will plot the latest BREF1 NIDS image. The file name convention uses the "bref1" as the extension (%e) in naming the file.

```
rad -if=nids_cref -cu=la -de=d -va=comp
```

This will plot the latest composite reflectivity image overlaid with the composite storm attributes.

### FILES

- **rad.cty** - the radar location database file
- **rad.var** - the variable menu and definitions for MDR display
- **nids.var** - the variable menu and definitions for NIDS storm attributes

### SEE ALSO

- [radcvf](#) - the MDR radar data decoding program

# RADCVT

## NAME

**radcv**t - MDR radar data conversion program

## SYNOPSIS

**radcv**t [*parameters...*] *filename*

## PARAMETERS

| Command Line         | Resource                  | Default           | Description  |
|----------------------|---------------------------|-------------------|--|
| <b>-h</b>            | <a href="#">help</a>      | No                | Lists basic help information.  |
| <b>-df=filename</b>  | <a href="#">default</a>   | .wxddef           | Sets the name of the resource file.  |
| <b>-na=name</b>      | <a href="#">name</a>      | radcvt            | Specifies the name used in resource file parsing.  |
| <b>-ba</b>           | <a href="#">batch</a>     | No                | Run program in batch mode  |
| <b>-me=level</b>     | <a href="#">message</a>   | <b>out2</b>       | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• decoded output - <b>out3a</b></li> <li>• MDR information and status - <b>out3c</b></li> <li>• not in database - <b>out3c</b></li> <li>• primary MDR information - <b>out4a</b></li> <li>• echo data - <b>out4c</b></li> </ul>   |
| <b>-fp=filepath</b>  | <a href="#">file_path</a> | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>  | <a href="#">data_path</a> | current directory | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>   | <a href="#">con_path</a>  | current directory | Specifies the location (path) of the output converted data files. This may be modified in the name convention file.  |
| <b>-nc=name_conv</b> | <a href="#">name_conv</a> | <b>name_conv</b>  | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-inp=input</b>    | <a href="#">input</a>     | <b>mdr</b>        | Specifies the input file type. The default is <b>mdr</b> . Possible values are: <ul style="list-style-type: none"> <li>• <b>mdr</b> - MDR or RCM radar data from converted file (Default). Name convention set to <b>rad_dat</b>.</li> <li>• <b>raw</b> - Raw MDR data from ingested radar file Name convention set to <b>rad_dat</b>.</li> <li>• <b>rcm</b> - RCM summary data. Name convention set to <b>rcm_dat</b>.</li> </ul> |
| <b>-if=in_file</b>   | <a href="#">in_file</a>   | <b>rad_dat</b>    | Specifies the input file name tag. The default is <b>rad_dat</b> or <b>rcm_dat</b> (RCM), but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-ou=output</b>    | <a href="#">output</a>    | <b>wxp</b>        | Specifies what type of output file is to be created: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII converted file. Name convention set to <b>rad_cvt_wxp</b>.</li> <li>• <b>cdf</b> - netCDF binary converted file. Name convention set to <b>rad_cvt_cdf</b>.</li> </ul>  |
| <b>-of=out_file</b>  | <a href="#">out_file</a>  | <b>rad_cvt</b>    | The <b>out_file</b> resource specifies the output file name tag. The default is <b>rad_cvt</b> but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |

|  |                            |   |  |
|--|----------------------------|---|--|
|  |                            |   | Also, can specifies what type of output file is to be created: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII converted file</li> <li>• <b>cdf</b> - netCDF binary converted file</li> </ul>  |
| <b>-pf=</b> <i>file_param</i>  | <a href="#">file_param</a> | User prompt<br>Batch: <b>use</b>                | This specifies how to treat files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul> |
| <b>-cf=</b> <i>cityfile</i>  | <a href="#">city file</a>  | <b>rad.cty</b>                                  | Specifies the name of the city database file used in finding station locations.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]                                  | <a href="#">current</a>    | None  | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=</b> <i>hour</i>  | <a href="#">hour</a>       | None  | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-pa=</b> <i>param</i><br>[, <i>param...</i> ]                         | <a href="#">parameter</a>  | None  | Extra parameters: <ul style="list-style-type: none"> <li>• <b>rawsum</b> - use the raw summary in file rather than deriving one from MDR reports</li> </ul>  |
| <i>filename</i> [# <i>seq</i> ]<br>[ <i>filename</i> [# <i>seq</i> ]...] | <a href="#">filename</a>   | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified.  |

## DESCRIPTION

The intent of the radar data conversion process is to decode either MDR (Manually Digitized Radar) or RCM (Radar Coded Messages) data for future plotting and to create a national radar summary based on radar site echo information.

### MDR Conversion

The program first decodes the MDR data for the following information:

- Radar site identifier,
- Report time,
- Report type, (area,line,no echoes,not available,etc.)
- Area coverage in percent,
- Type of precipitation, (rain,snow,thunderstorms,etc.)
- Intensity,
- Trend, (increasing or decreasing intensity)
- Location of precipitation,
- MDR echoes taken from local grid.

Then using the **rad.cty** file to determine the location of the radar site, a radar summary of the continental United States is constructed from the MDR echo data. This summary lies on a 120 (east-west) by 90 (north-south) grid that is then saved into the output converted file in a format compatible with the radar summary display program. The additional decoded data is saved to the output file for annotating the radar summary.

The **radcv**t program will first prompt the user for the input raw data file. This file should contain the MDR (SDUS) reports for a given time. It is recommended that these files be split up on hourly boundaries since this type of data is reported once an hour. The user may specify the input file either via the command line or through the [current](#)

resource. The input naming convention is specified by the **rad\_dat** tag but this can be changed with the [in\\_file](#) resource.

Once the input file is determined, the program determines the decode time from the filename specified or that of the **current** resource. Once the input file is set, the output converted file is opened for output. The name convention is determined from the **rad\_cvt** file convention tag. If the converted file exists, the user will be prompted whether to overwrite, append or use the existing file. The append option leaves the current data in the converted file and adds the new station data onto the end of the file. If the use option is specified, the converted file is not modified and the surface data conversion program exits.

There are two types of output data file types available. The ASCII wxp Flat file data type saves all available information in a columnar format which may be edited with a standard text editor. The netCDF data format is a binary format which may be produced by specifying **cdf** for the output file type.

### Raw MDR Summary

As part of the standard FOS feeds, the MDR radar summary is broadcast. If this MDR is desired over the one **radcvt** can produce, this summary can be saved to the output converted file. To select this, use **raw** for the **input** resource. This option will not decode MDR data and won't add additional MDR data to the converted file. In other words, no annotation is possible from the **rad** program.

### RCM Conversion

RCM decoding is very similar to MDR decoding. The RCM report is decoded into the following information:

- Site status
- Max echo tops
- Storm identifier
- Storm echo tops
- Storm movement
- Hail probability
- Tornado Vortex Signature location
- Mesocyclone location
- Echo data

The echo data is compiled into a radar summary on a 440x360 grid (4 times the MDR grid size) and saved to the output file. The additional decode storm attribute information is appended to the end of the converted file so that **rad** can annotate the plot with the storm attribute data.

**NOTE:** **radcvt** must be run before any radar plotting can be performed..

### EXAMPLES

```
radcvt -cu=la -pf=over -me=print
```

This converts the latest MDR data file and overwrites any existing file.

```
radcvt -inp=rcm -cu=la -pf=over
```

This converts the latest RCM data file and overwrites any existing file.

### FILES

- **rad.cty** - the radar site location database file

### SEE ALSO

- [rad](#) - the radar data plotting program

# RAWMATH

## NAME

**rawmath** - Mathematical manipulation program for raw files

## SYNOPSIS

**rawmath** [*parameters...*] *operations*

## PARAMETERS

| Command Line                            | Resource                   | Default                        | Description  |
|---|----------------------------|--------------------------------|--|
| <b>-h</b>                               | <a href="#">help</a>       | No                             | Lists basic help information.  |
| <b>-df=filename</b>                     | <a href="#">default</a>    | .wxpdef                        | Sets the name of the resource file.  |
| <b>-na=name</b>                         | <a href="#">name</a>       | rawmath                        | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                              | <a href="#">batch</a>      | No                             | Run program in batch mode  |
| <b>-me=level</b>                        | <a href="#">message</a>    | out2                           | Specifies level of messages to be displayed  |
| <b>-fp=filepath</b>                     | <a href="#">file_path</a>  | current directory              | Specifies location of database files. All raw files used in this program are assumed to be in the <b>file_path</b> directory.  |
| <b>-rp=rawpath</b>                      | <a href="#">raw_path</a>   | current directory              | Specifies the location/path of input and output raw files.   |
| <b>-of=out_file</b>                     | <a href="#">out_file</a>   | wxp                            | Specifies what type of output raw file the program will produce. The input file type is queried when the file is opened but the user may change the output file type. The options are: <ul style="list-style-type: none"> <li>• <b>wxp</b> - ASCII WXP raw file format</li> </ul>  |
| <b>-pf=file_param</b>                   | <a href="#">file_param</a> | prompt user, use in batch mode | Specifies how to deal with writing to files that already exist. Possible values are: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program sometimes terminates when this occurs. This is especially true for decoders which perform no other operation than to create this file.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> </ul>   |
| <b>-pa=param</b><br>[ <i>param...</i> ] | <a href="#">parameter</a>  | None                           | Extra parameters: <ul style="list-style-type: none"> <li>• <b>miss:#</b><br/>Specifies value to use for all missing stations in a file to file operation.</li> <li>• <b>noattr</b><br/>Specifies to remove attribute information when writing to the output file.</li> </ul>   |
| <i>command</i>                          | <a href="#">command</a>    | None                           | The command resource specifies the files and values to use in the mathematical operations along with the function to perform. The commands are of the following formats: <ul style="list-style-type: none"> <li>• <b>func</b> <i>F1 F2 ... = FO [Type] [Date]</i></li> <li>• <i>F1 oper F2 = FO [Type] [Date]</i></li> </ul> <p>The file specifications (<i>F1,F2</i>) can either be a filename or a number. The filename can be specified with an extension to define a specific grid within a multiple grid file:</p> <p><i>filename[-field]</i></p> |

|  |  |  |  |
|--|--|--|--|
|  |  |  | <p>The field specification can be a column number or a field identifier. The field identifier is a string that accompanies a column header in the raw file or a string that precedes the data such as HGIRG=5.24</p> <p><i>FO</i> is the specific output filename. All filenames are prepended with the <a href="#">grid_path</a> resource value unless a full path is specified.</p> <p><i>Type</i> and <i>Date</i> are optional strings used to modify the information header of the grid file. This is useful in making sure the resulting grid has the proper information since the math functions often misrepresent the final data type.</p> |
|--|--|--|--|

## DESCRIPTION

The grdmath program performs mathematical operation on gridpoint data. The functions and the filenames are interpreted from the command line as part of the **command** resource. There are two syntax's used.

### Algebraic Format

The first form uses an algebraic approach:

$$F1 \text{ oper } F2 = FO [Type] [Date]$$

where the operator (**oper**) is a simple algebraic operator:

- **add**
- **sub**
- **mul**
- **div**
- **mod** - modulus operator (remainder of A/B)
- **ge** - returns value if greater than or equal to second value, otherwise returns missing
- **le** - returns value if less than or equal to second value, otherwise returns missing

The values *F1* and *F2* represent either filenames or numeric values. The numeric values have the following syntax:

*#number*

The number listed can be any integer or floating point number. This option also allows the specification of the Coriolis force (**#cor**). This value is calculated for each grid point and added, subtracted, etc based on the math function.

The filenames have the following syntax:

*filename[-field]*

where *filename* is the name of the grid file to use. If a relative path is used, the value of the **raw\_path** is prepended to the filename. The filename can be "**std**" where standard input is used.

The extra information following the "-" defines which field to use within a raw file. The field specification can be a column number or a field identifier. The field identifier is a string that accompanies a column header in the raw file or a string that precedes the data such as HGIRG=5.24



```
sfcwx -re=us -pr=3 -va=temp -of=std -dr=none -cu=la-24 ) | \
rawmath - sub - = - "24 hour Temp change (F)" | \
mapplt -re=us -de=d -
```

This subtracts two raw files produced by sfcwx to produce a 24 hour temperature change. In the case, sfcwx is used to select station data for the US but instead of saving to file, the result is written to standard output and piped to rawmath. The resulting raw file is then displayed with mapplt.

In order to tally 24 hour precipitation data, rawmath can be used to add these values on a daily basis:

```
sfcwx 12061597 -va=r2 -pr=4 -of=prec15.raw -dr=none -re=us
sfcwx 12061697 -va=r2 -pr=4 -of=prec16.raw -dr=none -re=us
```

or

```
sfcwx -ho=12 -cu -va=r2 -pr=4 -of=prec17.raw -dr=none -re=us
```

will create raw files containing 24 hour precipitation data. To add them into a monthly database:

```
rawmath prec_jun.raw add prec15.raw = prec_jun.raw -pa=miss:0
```

or

```
rawmath sum prec???.raw = prec_jun.raw -pa=miss:0
```

The **miss:0** parameter specifies to keep stations missing in one but not both data files by substituting the value **0** for the missing station. Using the above example, if Chicago (ORD) did not report precipitation on the 15th, the ORD value from the monthly database would normally be discarded. To maintain ORD in the output file, **miss:0** must be specified. This will say to use the value listed (**miss:value**) to add to the value of ORD in the monthly database to put the resulting value in the output file. You can create the monthly database raw file with:

```
rawmath prec01.raw add #0 = prec_jun.raw "Monthly precip (in)" "June 1993"
```

This is a simple way to just copy the values from the prec01.raw file to the prec\_jun.raw file. To plot the results:

```
mapplt -fm=.2 -re=us -sc=.5 prec_jun.raw
```

To compare to normal:

```
rawmath prec_jun.raw div prec_norm_jun.raw = - |
rawmath - mul #100 = prec_normp_jun.raw "Percent of June normal prec"
```

To plot the results:

```
mapplt -fm=.0 -re=us -sc=.5 prec_normp_jun.raw
```

Rawmath can also develop time series with the "series" function":

```
rawmath series prec10.raw prec11.raw prec12.raw ... = - \
-id=LAF -pa=miss:0 -fm=%3.2f "Daily precip for LAF" "NOV-DEC 1996"
```

The series function allow data from several raw files to be incorporated into one raw file. The default is to give the IDs from the first file the number 1, the second file 2, and so on. The *ident* parameter is used to isolate LAF from each raw file. Again, the **miss:0** is used to put the value 0 in the output raw file even if LAF did not report a precipitation value for that that day. The **-fm=%3.2f** specifies to format the output in the raw file as %3.2f (in C). The result will be:

```
WXPRAW
```

```

NOV-DEC 1996
Daily precip for LAF (in)
1 0.00
2 0.00
3 0.00
4 0.35
5 0.67
6 0.64
...

```

To override the ID specification, add a user specified ID after the filename:

```

rawmath series prec10.raw:10_NOV prec11.raw:11_NOV prec12.raw:12_NOV \
...= - -id=LAF -pa=miss:0 -fm=3.2 "Daily precip for LAF" "NOV-DEC 1993"

```

The result is:

```

WXPRAW
NOV-DEC 1996
Daily precip for LAF
10_NOV 0.00
11_NOV 0.00
12_NOV 0.00
13_NOV 0.35
14_NOV 0.67
15_NOV 0.64
...

```

To plot the data using mapplt:

```

mapplt -pd=cat,0,0,3,2,30,2 -cod=cyan:w5 \
-pp=lalo:3:.2,bar:0,mark:none prec_LAF.raw

```

Rawmath has the capability of creating time series of SHEF data. For example, rawmath could be used to create a new raw file with:

```

rawmath series 12111996.shf-HGIRG:NOV19 12112096.shf-HGIRG:NOV20 \
12112196.shf-HGIRG:NOV21 12112296.shf-HGIRG:NOV22 = series.raw -id=LAFI3 \
-fm=%5.2f "SHEF-HGIRG for LAFI3"

```

```

WXPRAW
12Z 19 NOV 93
SHEF-HGIRG for LAFI3
NOV19 15.43
NOV20 13.75
NOV21 11.05
NOV22 9.39

```

The mapplt program will create a time-value plot:

```

mapplt -pd=cat,0,5,2,2,3,15 -pp=line,mark:1,lalo:1:1 -de=d series.raw

```

This uses the series function to extract **HGIRG** data from the SHEF raw files. This is labeled by day for the station **LAFI3**.

## FILES

- [Raw files](#)

## SEE ALSO

- [sfcwx](#)
- [upairwx](#)
- [fouswx](#)
- [shfcvt](#)
- [climat](#)
- [grid](#)

# SA\_PARSE

## NAME

**sa\_parse** - Surface data display program

## SYNOPSIS

**sa\_parse** [*parameters...*] *filename*

## PARAMETERS

| Command Line          | Resource                    | Default                     | Description   |
|-----------------------|-----------------------------|-----------------------------|---|
| <b>-h</b>             | <a href="#">help</a>        | No                          | Lists basic help information.   |
| <b>-df=filename</b>   | <a href="#">default</a>     | .wxpdef                     | Sets the name of the resource file.   |
| <b>-na=name</b>       | <a href="#">name</a>        | sa_parse                    | Specifies the name used in resource file parsing.   |
| <b>-ba</b>            | <a href="#">batch</a>       | No                          | Run program in batch mode   |
| <b>-me=level</b>      | <a href="#">message</a>     | out2                        | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>  |
| <b>-fp=filepath</b>   | <a href="#">file_path</a>   | current directory           | Specifies location of database files.   |
| <b>-dp=datapath</b>   | <a href="#">data_path</a>   | current directory           | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>    | <a href="#">con_path</a>    | current directory           | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.  |
| <b>-nc=name_conv</b>  | <a href="#">name_conv</a>   | name_conv                   | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>    | <a href="#">in_file</a>     | sfc_dat                     | Specifies the input file name tag. The default is <b>sfc_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well. If the input type is " <b>cvt</b> ", the program will use converted data. The input name tag will then be <b>sfc_cvt</b> unless otherwise specified. |
| <b>-cu=[hour]la</b>   | <a href="#">current</a>     | None                        | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=hour</b>       | <a href="#">hour</a>        | None                        | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh=num_hour</b>   | <a href="#">num_hour</a>    | 0                           | This specifies the number of hours that will be used. If this is not specified, a single hour will be parse. Otherwise a set of hours will be parsed.   |
| <b>-cf=cityfile</b>   | <a href="#">city_file</a>   | sao.cty                     | Specifies the name of the city database file used in finding station locations.   |
| <b>-pd=domain</b>     | <a href="#">plot_domain</a> | Not set. All stations used. | This specifies the plotting domain for station filtering. The projection, aspect ratio and the domain size are specified.   |
| <b>-re=region</b>     | <a href="#">region</a>      | Same as plot domain         | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-pr=prior</b>      | <a href="#">stat_prior</a>  | Not set. All stations used. | This specifies the lowest priority to use in station filtering. Priorities range from 1 to 7.   |
| <b>-id=identifier</b> | <a href="#">identifier</a>  | User prompt                 | Specifies the station to use in the program. A list of stations separated by commas can be specified. To get all stations, use the keyword " <b>all</b> ".<br>Also, using keywords will parse for more than just specific stations. This requires the city file to be read so use the " <b>city</b> "   |

|                       |                           |   |  |
|-----------------------|---------------------------|---|--|
|                       |                           |   | parameter. For example, " <b>st=state</b> " will parse for all stations within a state. This uses the <b>sao.cty</b> file to search for IDs within a state.  |
| <b>-pa=param</b>      | <a href="#">parameter</a> | None  | Specifies additional plotting parameters. See the <a href="#">parameter</a> resource for more details. Some possibilities are: <ul style="list-style-type: none"> <li>• <b>dec</b> -- decode raw surface data prior to output</li> <li>• <b>raw</b> -- print raw undecoded surface data</li> <li>• <b>pln</b> -- single line output for decoded printout</li> <li>• <b>plns</b> -- double line output for decoded printout (default)</li> <li>• <b>psyn</b> -- use simple synoptic format for decoded printout</li> <li>• <b>psimp</b> -- single line output for general use</li> <li>• <b>pfull</b> -- multiline output for decoded printout</li> <li>• <b>pvar</b> -- uses <b>variable</b> resource to create table of station data</li> <li>• <b>blank</b> -- display stations not reporting</li> <li>• <b>city</b> -- use a city database to sort station data into a particular order</li> <li>• <b>cvt</b> -- parse a decoded file rather than a raw file</li> <li>• <b>header-1</b> -- print a column header only once</li> <li>• <b>header-e</b> -- print a column header for every station</li> </ul> |
| <i>filename[#seq]</i> | <a href="#">filename</a>  | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the surface data file to be used. An optional sequence number can be added to designate the time for non-WXP files.  |

## DESCRIPTION

This program displays surface data in a tabular format. The input to the program is either a raw ingested surface data file or a converted surface data file produced by the [sacvt](#) program.

The programs starts off by prompting the user for input data file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **sfc\_dat** tag (or **sfc\_cvt** if "cvt" data is requested) but this can be changed with the [in\\_file](#) resource.

Next, the user enters a station identifier. The stations are entered by their 3/4 letter IDs. Once the station is selected, the program prints out surface data.

## Text Output

There are 4 formats for surface data output:

- **raw** - this prints the raw undecoded data (second line will be indented):

```
KBIV 070242Z AUTO 13006KT 5SM -RA BR SCT015 15/15 A2995 RMK AO2 P0002 TSNO
```

- **pln** - this prints a single line of decoded output:

```
ID  TIME  T  TD  RH  DIR  SPD  GST  ALT  SLP  VIS  CIL  COV  WX  MAX  MIN  PR6  PR24  SC
KLNP 0240 52 43 71 150  4    002    10    CLR
K4BK 2347 69 52 54 310 10    114  30    CLR    72  59
```

- **plns** - this is a 2 line output of decoded output plus an optional third line for comments:

```
ID Tp Time  Vis  PWx  SLPres  T  Td  WD-WS  WG  Alt  PT  Max  Min
  Cht CCv CHT CCv CHT CCv CHT CCv CHT CCv  L:M:H  Rn3  Rn6 Rn12 Rn24 SC
```

```

PATL SP 0335 10.0          64 51 290- 4    29.58
    43 SCT                  :  :
    COMMENT: AO1
PAEH SP 0335 10.0          49 43 340- 4    29.66
    6 SCT                   :  :
    COMMENT: AO1

```

- **psyn** - this prints out a single line of decoded output setup for synoptic data:

```

ID   Cht Cv  Vis  PWx  SLPres  T  Td  WD-WS WG  Te  PT   Rn6 Rn24 SC L:M:H
KSLC 250 F 15.0      1004.7 68 58 280- 7

```

- **psimp** - this prints out a single line of decoded data more appropriate for general use:

```

ID   Temp  Dew  Winds      Press  Clouds  Prec  Weather
      F    F   deg  knt      mb 100ft cov  in
MGGT 64    63  360 at 14  ----- ---      ---- heavy drizzle
MGGT 64    63  360 at 14  ----- ---      ---- heavy drizzle
PAWR 55    46   90 at 3  1024.0 ---      ----
PASD 46    39   50 at 5  ----- ---      ----

```

- **pfull** - this is a multi-line output per station more appropriate for general use:

```

Data for: KHOU
-----
Report time:      0450 Z
Temperature:      82 F = 27.5 C
Dewpoint:        78 F = 25.3 C
Winds:           160 deg at 6 knt
Altimeter setting: 29.85 in Hg
Sea level pressure: 1010.8 mb
Clouds:level 1:  1500 feet few
               level 2: 2500 feet scattered
Visibility:      10.00 miles
Heat index:      89.5 F

```

- **pvar** - use the **variable** resource to create the table of station data (see below).

Column headings include:

- **ID** - Station identifier
- **Tp** - Type of report (SA,RS,SP)
- **Time** - Time of the report in GMT
- **Vis** - Visibility in miles
- **PWx** - Present weather (RW-F for example)
- **SLP/SLPres** - Sea level pressure in mb
- **T** - Temperature in F
- **Td** - Dewpoint in F
- **DIR/WD** - Wind direction
- **SPD/WS** - Wind speed (separated from direction by a hyphen)
- **GST/WG** - Gust data (left blank if none reported)
- **Alt** - Altimeter setting in inches of Hg
- **PT** - Pressure tendency in standard coded form (310 for example)
- **PR6/Rn6** - 3 or 6 hour precipitation in inches (separated from PT by a colon)
- **PR24/Rn24** - 24 hour precipitation in inches
- **Te** - Extreme temperature in F
- **Max** - maximum temperature in F
- **Min** - minimum temperature in F

- **SC** - Snow cover in inches
- **CIL/CHt** - Cloud ceiling height data
- **COV/Cv/CCv** - Cloud coverage data for each cloud level as text "SCT" or eighths "1/8"
- **L, M, H** - Low, middle and high cloud type information (separated by colons)

### Printing using Variable Interface

If the print type is set to **pvar**, the **variable** resource is used to determine which variables will be printed in the table. There is a **sa\_parse.var** file for setting up complicated tables and simplifying program use. The possible formatting includes the units and output C format (**wetblb:F:%.1f**). Also, time can be specified for use with precipitation (**anal-6acc::prec::%.2f**). The data is displayed in 7 character wide columns. The output for "city,temp,dewp,rhum,wetblb:F:%.1f" would look like:

```

      city   temp   dewp   rhum  wetblb
                                F
KTHV         83    66     57   71.1
KIPT         85    64     50   70.4
KAGC         84    61     45   68.4
KPIT         82    61     49   67.9
KPHL         84    73     70   76.0
KFKL         81    61     51   67.4
KERI         82    65     56   70.2
KAFJ         82    61     48   67.9
KSEG         85    63     48   69.9
...

```

Here is a list of the possible variables:

### Internal Variables

| Variable                 | Description and units  |
|--------------------------|--|
| <b>city</b> or <b>id</b> | Station ID   |
| <b>otime</b>             | Observation time   |
| <b>temp</b>              | Temperature (F)  |
| <b>dewp</b>              | Dewpoint (F)   |
| <b>dewd</b>              | Dewpoint depression (F)  |
| <b>wdir</b>              | Wind direction (deg)   |
| <b>wspd</b>              | Wind speed (knt)   |
| <b>wgst</b>              | Wind gust (knt)  |
| <b>alt</b>               | Altimeter setting (in_Hg)  |
| <b>pres</b>              | Sea level pressure (mb) use level parameter to get station/surface pressure  |
| <b>pralt</b>             | Sea level pressure or altimeter setting if sea level pressure is missing (mb)  |
| <b>elev</b>              | Station elevation (m)  |
| <b>wx</b>                | Present weather (string)   |
| <b>vis</b>               | Visibility (mi)  |
| <b>cldcl</b>             | Cloud ceiling (100 ft)   |
| <b>cldcv</b>             | Cloud cover (cloud string)   |
| <b>ptend</b>             | Pressure tendency (mb, trend is a string)  |
| <b>prec</b>              | Precipitation (in) use time parameter to specify which type (3,6,12,24 hour)   |
| <b>extt</b>              | Extreme temperature (F) uses maximum if available, else minimum.<br>Use time parameter to specify which type (6,24 hour) |
| <b>maxt</b>              | Maximum temperature (F) use time parameter to specify which type (6,24 hour)   |
| <b>mint</b>              | Minimum temperature (F) use time parameter to specify which type (6,24 hour)   |
| <b>snwdp</b>             | Snow depth (in)  |
| <b>lcl</b>               | Low cloud type (string)  |

|               |  |
|---------------|--|
| <b>mcl</b>    | Middle cloud type (string)                             |
| <b>hcl</b>    | High cloud type (string)                               |
| <b>solar</b>  | Equivalent solar radiation (min)                       |
| <b>sst</b>    | Sea surface temperature (C)                            |
| <b>wavper</b> | Wave period (sec)                                      |
| <b>wavhgt</b> | Wave height (m)  |
| <b>vpres</b>  | Vapor pressure (mb)                                    |
| <b>svpres</b> | Saturation vapor pressure (mb)                         |
| <b>whilt</b>  | Wind chill temperature (K)                             |
| <b>wchilf</b> | Wind chill factor (W/m^2)                              |
| <b>heat</b>   | Heat index (K)   |
| <b>humit</b>  | Humiture (F)   |
| <b>mrat</b>   | Mixing ratio (g/kg)                                    |
| <b>smrat</b>  | Saturation mixing ration (g/kg)                        |
| <b>shum</b>   | Specific humidity (g/kg)                               |
| <b>rhum</b>   | Relative humidity (%)                                  |
| <b>wetblb</b> | Wetbulb temperature (K)                                |
| <b>thetaw</b> | Wetbulb potential temperature (K)                      |
| <b>thetae</b> | Equivalent potential temperature (K)                   |
| <b>thetav</b> | Virtual potential temperature (K)                      |
| <b>theta</b>  | Potential temperature (K)                              |
| <b>vtemp</b>  | Virtual temperature (K)                                |
| <b>fr</b>     | Flight rules (number and string: 0=VFR, 1=MVFR, 2=IFR) |

## EXAMPLES

```
sa_parse -cu=la -id=KIND
KIND 020756Z 11003KT 9SM CLR 06/06 A3013 RMK AO2 SLP203 T00560056      $
```

This displays surface data for KIND.

```
sa_parse -cu=la -id=KIND -if=cvt
ID  TIME  T  TD  RH  DIR  SPD  GST  ALT  SLP  VIS  CIL  COV  WX  MAX  MIN  PR6  PR24  SC
KIND 0756  42  42 100 110   3   013 203   9   CLR
```

This displays a user specified list of variables:

```
sa_parse -cu=la -if=cvt -id=ST=PA -pa=city,pvar \
  -va=city,temp,dewp,rhum,wetblb:F:%.1f
  city  temp  dewp  rhum  wetblb
                                     F
KTHV      83    66    57    71.1
KIPT      85    64    50    70.4
KAGC      84    61    45    68.4
KPIT      82    61    49    67.9
KPHL      84    73    70    76.0
KFKL      81    61    51    67.4
KERI      82    65    56    70.2
KAFJ      82    61    48    67.9
KSEG      85    63    48    69.9
...
```

## FILES

- **sao.cty** - the surface station location database file

## SEE ALSO

- [sacvt](#) - the surface data decoding program

# SACVT

## NAME

**sacvt** - Surface METAR/SAO data conversion program

## SYNOPSIS

**sacvt** [*parameters...*] *filename*

## PARAMETERS

| Command Line          | Resource                   | Default                          | Description   |
|-----------------------|----------------------------|----------------------------------|---|
| <b>-h</b>             | <a href="#">help</a>       | No                               | Lists basic help information.   |
| <b>-df=filename</b>   | <a href="#">default</a>    | .wxpdef                          | Sets the name of the resource file.   |
| <b>-na=name</b>       | <a href="#">name</a>       | sacvt                            | Specifies the name used in resource file parsing.   |
| <b>-ba</b>            | <a href="#">batch</a>      | No                               | Run program in batch mode   |
| <b>-me=level</b>      | <a href="#">message</a>    | out2                             | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>decoded output - <b>out3a</b></li> <li>raw data and format - <b>out3c</b></li> <li>not in database - <b>out3c</b></li> <li>remark decode status - <b>out4c</b></li> </ul>  |
| <b>-fp=filepath</b>   | <a href="#">file_path</a>  | current directory                | Specifies location of database files.   |
| <b>-dp=datapath</b>   | <a href="#">data_path</a>  | current directory                | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>    | <a href="#">con_path</a>   | current directory                | Specifies the location (path) of the output converted data files. This may be modified in the name convention file.   |
| <b>-nc=name_conv</b>  | <a href="#">name_conv</a>  | name_conv                        | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>    | <a href="#">in_file</a>    | sfc_dat                          | Specifies the input file name tag. The default is <b>sfc_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-ou=output</b>     | <a href="#">output</a>     | wxp                              | Specifies what type of output file is to be created: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII converted file. Name convention set to <b>sfc_cvt_wxp</b>.</li> <li><b>cdf</b> - netCDF binary converted file. Name convention set to <b>sfc_cvt_cdf</b>.</li> </ul>   |
| <b>-of=out_file</b>   | <a href="#">out_file</a>   | sfc_cvt                          | The <b>out_file</b> resource specifies the output file name tag. The default is <b>sfc_cvt</b> but it can be modified to any value in the filename convention file. A full name convention can be specified as well.<br><br>Also, can specify what type of output file is to be created: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII converted file</li> <li><b>cdf</b> - netCDF binary converted file</li> </ul> |
| <b>-pf=file_param</b> | <a href="#">file_param</a> | User prompt<br>Batch: <b>use</b> | This specifies how to treat files that exist: <ul style="list-style-type: none"> <li><b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li><b>Over</b> - overwrite the existing file</li> <li><b>App</b> - append the information onto the end of the existing file</li> </ul>   |

|  |                             |   |   |
|--|-----------------------------|---|---|
|  |                             |   | <ul style="list-style-type: none"> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul>   |
| <b>-cf</b> = <i>cityfile</i>   | <a href="#">city file</a>   | None  | Specifies the name of the city database file used in finding station locations.   |
| <b>-cu</b> =[ <i>hour</i> ] <b>la</b>                                    | <a href="#">current</a>     | None  | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho</b> = <i>hour</i>   | <a href="#">hour</a>        | None  | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh</b> = <i>num_hour</i>   | <a href="#">num hour</a>    | <b>0</b>  | This specifies the number of hours that will be searched for hourly data.   |
| <b>-dh</b> = <i>decode_hour</i>  | <a href="#">decode hour</a> | None  | Specifies a specific decode time when using more than one file.   |
| <b>-id</b> = <i>identifier</i>   | <a href="#">identifier</a>  | None  | Used to decode a specific observation. If not specified, all stations will be decoded.  |
| <b>-pa</b> = <i>param</i><br>[, <i>param...</i> ]                        | <a href="#">parameter</a>   | None  | Extra parameters: <ul style="list-style-type: none"> <li>• <b>time</b> - print the time in the output file. (default)</li> <li>• <b>no_loc</b> - no location in the input file, use city database.</li> <li>• <b>spec</b> - save specials to output file (default)</li> <li>• <b>nospec</b> - does not save specials.</li> <li>• <b>pln</b> - print single line report</li> <li>• <b>plns</b> - print the 2 line report</li> <li>• <b>psyn</b> - print using synoptic print format.</li> <li>• <b>pfull</b> - print using the full multiline report</li> <li>• <b>range:pre:post</b> - specifies time range for valid data.(default: -15:10)</li> <li>• <b>all</b> - save all data to output file.</li> <li>• <b>dup</b> - remove duplicate reports prior to saving to file</li> <li>• <b>update</b> - remove duplicates and update existing reports with corrections.</li> </ul> |
| <i>filename</i> [# <i>seq</i> ]<br>[ <i>filename</i> [# <i>seq</i> ]...] | <a href="#">filename</a>    | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified.   |

## DESCRIPTION

The intent of the surface data conversion process is to combine all the surface data (SAO, METAR, Synopt, Buoy, Dribu, CMAN) into one format so that a single program can plot all the data. The surface conversion process is split up into two programs **sacvt** (METAR/SAO) and **smcvt** (Synoptic, Buoy, CMAN). **sacvt** program handles hourly data in either METAR or SAO formats. All stations for a particular report time are decoded and put into an output converted data file.

The sacvt program will first prompt the user for the input raw data file. This file should contain the METAR and SAO reports for a given time. It is recommended that these files be split up on hourly boundaries since this type of data is reported once an hour. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **sfc\_dat** tag but this can be changed with the [in\\_file](#) resource. If the data stretches over an hour, the [num\\_hour](#) resource can be specified to search over multiple files.

Once the input file is determined, the program determines the decode time from the filename specified or that of the **current** resource. If decode time is different from that of the specified file, the [decode\\_hour](#) resource must be specified. For example, if the decode time is 12Z and the range of files is from 10-14Z, the current hour would be 14Z, the number of hours would be -4 and the decode hour would be 12. This resource can also be used to set

decode time range limits which is useful if decoding data from off times such as :20 after the hour. Once the decode time is set, the output converted file is opened for output. The name convention is determined from the **syn\_cvt** file convention tag. If the converted file exists, the user will be prompted whether to overwrite, append or use the existing file. The append option leaves the current data in the converted file and adds the new station data onto the end of the file. If the use option is specified, the converted file is not modified and the surface data conversion program exits.

The data are parsed for individual station reports which are then passed through the METAR/SAO decode module. Once decoded, the output is saved to the output converted file in the order they were decoded. This is not true if the **dup** or **update** options are specified. The remove duplicate (**dup**) option removes duplicate reports from the output converted file. The update station (**update**) option not only removes duplicate stations but updates station data with corrections and amendments.

The conversion process is silent. By default, none of the output is displayed to the screen as an attempt to speed up the conversion process. Decoded output can be displayed if the **out3a** message level is set:

```

ID   TIME   T  TD  RH DIR SPD GST ALT  SLP VIS CIL COV WX  MAX MIN  PR6 PR24 SC
Processing file: /rainbow/data/ddplus/03060797.sao
KLNP 0240  52 43  71 150  4    002    10    CLR
K4BK 2347  69 52  54 310 10    114    30    CLR    72  59
CYZY 0243  50 43  76 230  7  21 976    9  38 FEW
PABA 0235  36 32  86 100 11    975    10    CLR
KBIV 0242  59 59 100 130  6    995    5  15 SCT R-
KCAE 0242  61 55  82  20  8    995    10  23 OVC
PACV 0241  50 50 100 160  6    965 041  5  8 OVC R-
KPSX 0240  79 72  79 150 10    978    7 200 OVC
KJAX 0243  64 61  88 350  6    986    10  12 OVC
KCRP 0243  73 66  78 310 12  18 977    10  19 OVC TR-
WRA  0243  37 36  93  40  22    182
WEF  0242  39 39 100    14    212
CYQX 0242    10  8    0.8  2 OVC
WAF  0200  45 41  87  30  5
WRA  0243  37 36  93  40  22    182
M
```

If the raw information is to be displayed, use the **out3** message level:

```

ID   TIME   T  TD  RH DIR SPD GST ALT  SLP VIS CIL COV WX  MAX MIN  PR6 PR24 SC
Processing file: /rainbow/data/ddplus/03060797.sao
0:2:243:KLNP:KLNP 070240Z AUTO 15004KT 10SM CLR 11/06 A3002 RMK AO1
KLNP 0240  52 43  71 150  4    002    10    CLR
0:2:0:K4BK:K4BK 062347Z COR 31010KT 30SM SKC 21/11 RMK SLP114 NOSPECI    T0207
0111 10223 20152 57010
K4BK 2347  69 52  54 310 10    114    30    CLR    72  59
1:5:243:CYZY:SPECI CYZY 070243Z AUTO 23007G21KT 9SM FEW038 FEW048 FEW057 10/06 A
2976 RMK PRESRR
CYZY 0243  50 43  76 230  7  21 976    9  38 FEW
0:5:243:PABA:PABA 070235Z AUTO 10011KT 10SM CLR 02/00 A2975 RMK AO1    T002200
01
PABA 0235  36 32  86 100 11    975    10    CLR
0:5:243:KBIV:KBIV 070242Z AUTO 13006KT 5SM -RA BR SCT015 15/15 A2995 RMK AO2
P0002 TSNO
KBIV 0242  59 59 100 130  6    995    5  15 SCT R-
0:5:243:KCAE:KCAE 070242Z 02008KT 10SM BKN023 OVC031 16/13 A2995 RMK AO2
```

The raw report is preceded by 3 numbers which denote the location (0=US, 1=Canada, 2=Mexico), data type (1=SA, 2=METAR, 3=Synop, 4=SP, 5=SPECI, 8=Correction, 16=Automated, 32=ASOS) and the time of observation. For more information on output formats, see the [text output section of the users guide](#).

There are two types of output data file types available. The ASCII wxp Flat file data type saves all available information in a columnar format which may be edited with a standard text editor. The netCDF data format is a binary format which may be produced by specifying **cdf** for the output file type.

**NOTE:** **sacvt** must be run before any surface plotting and gridding programs may be run.

**NOTE:** **sacvt** produces the same converted file name as does **sacvt**. If the same hour is converted by **sacvt**, the program will prompt the user for overwrite or append.

## EXAMPLES

```
sacvt -cu=la -pf=over -me=print -pa=update
```

This converts the latest hour data file and overwrites any existing file. The station data is updated prior to saving the data to file.

## FILES

- **sao.cty** - the surface station location database file

## SEE ALSO

- [smcvt](#) - the synoptic data decoding program
- [sfcwx](#) - the surface data plotting program
- [sfccalc](#) - the surface data contouring program
- [statlog](#) - the surface meteogram plotting program

# SCOUR

## NAME

**scour** - File deletion program

## SYNOPSIS

**scour** [*options...*] [*path...*] [*limit...*]

## PARAMETERS

| Command Line | Description   |
|--------------|---|
| <b>-h</b>    | Lists help information.   |
| <b>-list</b> | Lists the files in the directory  |
| <b>-verb</b> | Prints information (default: no output)   |
| <b>+name</b> | Sorts files in alphabetical order (A-Z)   |
| <b>-name</b> | Sorts files in reverse alphabetical order (Z-A)   |
| <b>+size</b> | Sorts files in size order (0-9)   |
| <b>-size</b> | Sorts files in reverse size order (9-0)   |
| <b>+date</b> | Sorts files in age order (jan-dec)  |
| <b>-date</b> | Sorts files in reverse age order (dec-jan)  |
| <i>path</i>  | Path to scour (default: current directory)  |
| <i>limit</i> | Limit of scour (default: no scour performed) <ul style="list-style-type: none"> <li>• <i>size</i> - size in bytes to scour to</li> <li>• <b>ksize</b>=<i>size</i> - size in kilobytes to scour to</li> <li>• <b>msize</b>=<i>size</i> - size in megabytes to scour to</li> <li>• <b>age</b>=<i>age</i> - age in days to scour to</li> </ul> |

## DESCRIPTION

This program is used to delete old data files from the ingested data directories to prevent the hard disk from filling up. This program can be used to view directory information if the limit is not specified on the command line. If a limit is specified, the program will delete the oldest files until the limit is reached either by reducing the size of the directory to a specified byte count or removing all files older than a certain age. To insure scour is run, it is recommended that it be installed into the cron list for the user that is running the ingester (to prevent permission conflicts). Here is a sample crontab listing:

```
35 * * * * /home/wxp/bin/scour -verb /home/wxp/data msize=300
36 * * * * /home/wxp/bin/scour -verb /home/wxp/text msize=150
37 * * * * /home/wxp/bin/scour -verb /home/wxp/model msize=500
38 * * * * /home/wxp/bin/scour -verb /home/wxp/convert msize=25
38 * * * * /home/wxp/bin/scour -verb /home/wxp/sat msize=500
```

## EXAMPLES

```
scour /usr/wxp/data msize=400
```

This will delete the oldest files in the directory until the size of the directory is 400 megabytes.

```
scour /usr/wxp/convert age=10
```

This will remove all files which are greater than 10 days old.

## SEE ALSO

- [ingest](#) - WXP ingest program

# SFCCALC

## NAME

**sfccalc** - Surface data contouring program

## SYNOPSIS

**sfccalc** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description  |
|------------------------------|----------------------------|-------------------|--|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.  |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxddef           | Sets the name of the resource file.  |
| <b>-na=name</b>              | <a href="#">name</a>       | sfccalc           | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode  |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>gridding information - <b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | sfc_cvt           | Specifies the input file name tag. The default is <b>sfc_cvt</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of raw or grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII grid or raw file</li> <li><b>wxpb</b> - WXP binary grid file</li> <li><b>raw</b> - WXP ASCII raw file</li> <li><b>cdf</b> - NetCDF binary grid file</li> </ul> Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>sfc_raw</b> or <b>sfc_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type:<br><br>wxp:sfc_grd |

|   |                              |  |   |
|---|------------------------------|--|---|
|   |                              |  | A full filename can be specified which can contain name convention tags.  |
| <b>-pf=</b> <i>file_param</i>   | <a href="#">file_param</a>   | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul> |
| <b>-cf=</b> <i>cityfile</i>   | <a href="#">city file</a>    | <b>sao.cty</b>   | Specifies the name of the city database file used in finding station locations.   |
| <b>-mf=</b> <i>mapfile</i> [,...]   | <a href="#">map file</a>     | <b>wxp.map</b> or<br><b>cont.dat</b> ,<br><b>country.dat</b> for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]   | <a href="#">current</a>      | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=</b> <i>hour</i>   | <a href="#">hour</a>         | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh=</b> <i>num_hour</i>   | <a href="#">num hour</a>     | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.   |
| <b>-id=</b> <i>identifier</i>   | <a href="#">identifier</a>   | None   | Used to plot a specific observation. If not specified, all stations will be plotted.  |
| <b>-pd=</b> <i>domain</i>   | <a href="#">plot domain</a>  | User prompt<br>Batch: first<br>domain in<br><b>wxp.reg</b> file.                                     | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-gd=</b> <i>grid_domain</i>  | <a href="#">grid domain</a>  | Same as plot<br>domain   | This specifies the domain to use when fitting data to a grid. If this is not specified, it is the <b>plot_domain</b> .  |
| <b>-re=</b> <i>region</i>   | <a href="#">region</a>       | Same as plot<br>domain   | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-pr=</b> <i>prior</i>  | <a href="#">stat prior</a>   | Depends on<br>domain size  | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.  |
| <b>-va=</b> <i>variable</i>   | <a href="#">variable</a>     | User prompt<br>Batch: first<br>variable listed in<br>the <b>sfccalc.var</b><br>file                  | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.  |
| <b>-oa=</b> <i>smooth</i><br>[, <i>radinf</i> ][, <i>passes</i> ]<br>[, <i>converg</i> ]<br>[, <i>min_sta</i> ] | <a href="#">object param</a> | <b>1,3,1,,2,1</b>  | Specifies the objective analysis parameters. There are 5 parameters including smoothing, radius of influence, number of passes, convergence parameter and minimum number of stations.   |
| <b>-pl=</b> <i>plot_type</i>  | <a href="#">plot type</a>    | User prompt<br>Batch: fill<br>contours or<br>vectors   | Specifies the type of output plot. By default, the user is prompted for the plot type. To produce a contour or vector plot, use the following plot types:<br><b>Contours:</b> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> </ul>  |

|  |                              |   |   |
|--|------------------------------|---|---|
|  |                              |   | <ul style="list-style-type: none"> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <b>Vectors:</b> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>                | <a href="#">plot_scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>                      | <a href="#">plot_format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.  |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot_param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>              | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>                 | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.   |
| <b>-in=interval</b>                    | <a href="#">con_interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b>            | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.  |
| <b>-cb=base</b>                        | <a href="#">con_base</a>     | Lowest contour<br>value, maximum<br>wind speed                          | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.  |
| <b>-fl=font_list</b>                   | <a href="#">font_list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>                 | <a href="#">color_table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>               | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the text color for labels above and below the plot.   |
| <b>-cod=color[:attr]</b>               | <a href="#">color_data</a>   | <b>green</b>  | Specifies the color for plotted data.   |
| <b>-com=color[:attr]</b>               | <a href="#">color_map</a>    | <b>red</b>  | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>              | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.  |
| <b>-cowl=color[:attr]</b>              | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for wind barbs.   |
| <b>-cocd=color[:attr]</b>              | <a href="#">color_cloud</a>  | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for cloud cover symbols.  |
| <b>-cowx=color[:attr]</b>              | <a href="#">color_wx</a>     | <b>yellow</b><br><b>brown</b> for<br>inverse plots                      | Specifies the color for weather symbols   |
| <b>-cocm=color[:attr]</b>              | <a href="#">color_cmap</a>   | <b>black</b>  | Specifies the color for color fill contour maps.  |
| <b>-coco=color[:attr]</b>              | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for line contours.  |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color_fill</a>   | A predefined<br>color set based<br>on the <b>wxp.clr</b><br>color table | Specifies the color fill contour color set. See the <a href="#">color_fill</a> resource for more details.   |

|   |                             |   |  |
|---|-----------------------------|---|--|
| <b>-cola=</b> <i>color[:attr]</i>                   | <a href="#">color_label</a> | <b>white</b><br><b>black</b> for inverse plots. | Specifies the line contour label color.  |
| <b>-cocl=</b> <i>color[:attr]</i>                   | <a href="#">color_label</a> | <b>black</b>                                    | Specifies the color fill contour label color.  |
| <b>-de=</b> <i>device[,...]</i>                     | <a href="#">device</a>      | User prompt<br>Batch: <b>display</b>            | Specifies the output device.   |
| <b>-ti=</b> <i>title</i>                            | <a href="#">title</a>       | Program name                                    | Specifies the window title.  |
| <b>-ic</b>  | <a href="#">icon</a>        | <b>Off</b>                                      | Specifies whether to open window in iconified mode.  |
| <b>-bg=</b> <i>color</i>                            | <a href="#">background</a>  | <b>black</b><br><b>white</b> for inverse plots  | Specifies the background color of the window.  |
| <b>-ge=</b> <i>geometry</i>                         | <a href="#">geometry</a>    | <b>640x480</b>                                  | Specifies the window and paper geometry.   |
| <i>filename[#seq]</i><br><i>[filename[#seq]...]</i> | <a href="#">filename</a>    | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the surface converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified to generate a loop. |

## DESCRIPTION

This program contours surface data on a background map. The input to the program is a surface converted file produced by either the [sacvt](#) or [smcvt](#) programs. Then, based on a plot domain and a variable selected by the user, a plot is generated. Overlay plots, plots and loops can also be generated if the appropriate options are set.

The programs starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the [sfc\\_cvt](#) tag but this can be changed with the [in\\_file](#) resource. To generate a loop, the [num\\_hour](#) resource can be specified. This will tell the program to loop through this number of hours.

Next, the program prompts the user for the plot domain. This is done through a region menu which has predefined regions available. If needed a full plot domain can be specified to get a precise domain. The domain can be interactively changed using the mouse once the plot has been generated.

After the region has been set, the city database is read in and a cross-reference table is setup. Since converted data is not saved with location information, the program will refer to this table to location each station in the input file.

The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the [sfccalc.var](#) file. The internal variables are:

### Internal Variables

| Variable                                   | Description and units                | Plots   |
|--|--------------------------------------|---|
| <b>city</b>                                | All stations in the data file        | string  |
| <b>temp</b>                                | Temperature (F)                      | value   |
| <b>ttd</b>                                 | Temperature and dewpoint (F)         | value <sub>0</sub> =temp,<br>value <sub>1</sub> =dewp |
| <b>dewp</b>                                | Dewpoint (F)                         | value   |
| <b>dewd</b>                                | Dewpoint depression (F)              | value   |
| <b>wdir</b>                                | Wind direction (deg)                 | value   |
| <b>wspd</b>                                | Wind speed (knt)                     | value   |
| <b>wgst</b>                                | Wind gust (knt)                      | value   |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)       | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd   |
| <b>uwnd</b>                                | U wind component - EW relative (m/s) | value   |
| <b>vwnd</b>                                | V wind component - NS relative (m/s) | value   |

|               |   |  |
|---------------|---|--|
| <b>uwndg</b>  | U wind component - grid relative (m/s)  | value  |
| <b>vwndg</b>  | V wind component - grid relative (m/s)  | value  |
| <b>alt</b>    | Altimeter setting (in_Hg)   | value  |
| <b>pres</b>   | Sea level pressure (mb) use level parameter to get station/surface pressure   | value  |
| <b>pralt</b>  | Sea level pressure or altimeter setting if sea level pressure is missing (mb) | value  |
| <b>elev</b>   | Station elevation (m)   | value  |
| <b>wx</b>     | Present weather (SAO string)  | string   |
| <b>vis</b>    | Visibility (mi)   | value, formatted string  |
| <b>clld</b>   | Cloud ceiling (100 ft)  | value, formatted string  |
| <b>clldc</b>  | Cloud ceiling code (from synoptic code 0-9)                                   | value  |
| <b>cldev</b>  | Cloud cover (cloud string)  | value=%, string  |
| <b>ptend</b>  | Pressure tendency (mb, trend is a string)                                     | value, string=trend code   |
| <b>prec</b>   | Precipitation (in) use time parameter to specify which type (3,6,12,24 hour)  | value  |
| <b>extt</b>   | Extreme temperature (F) uses maximum if available, else minimum               | value  |
| <b>maxt</b>   | Maximum temperature (F) use time parameter to specify which type (6,24 hour)  | value  |
| <b>mint</b>   | Minimum temperature (F) use time parameter to specify which type (6,24 hour)  | value  |
| <b>snwdp</b>  | Snow depth (in)   | value  |
| <b>nlcd</b>   | Amount of low clouds (code string)  | value=%, string=code   |
| <b>lcl</b>    | Low cloud type (string)   | string   |
| <b>mcl</b>    | Middle cloud type (string)  | string   |
| <b>hcl</b>    | High cloud type (string)  | string   |
| <b>solar</b>  | Equivalent solar radiation (min)  | value  |
| <b>sst</b>    | Sea surface temperature (C)   | value  |
| <b>wavper</b> | Wave period (sec)   | value  |
| <b>wavhgt</b> | Wave height (m)   | value  |
| <b>vpres</b>  | Vapor pressure (mb)   | value  |
| <b>svpres</b> | Saturation vapor pressure (mb)  | value  |
| <b>wchilt</b> | Wind chill temperature (K)  | value  |
| <b>wchilf</b> | Wind chill factor (W/m^2)   | value  |
| <b>heat</b>   | Heat index (K)  | value  |
| <b>humit</b>  | Humiture (F)  | value  |
| <b>mrat</b>   | Mixing ratio (g/kg)   | value  |
| <b>smrat</b>  | Saturation mixing ration (g/kg)   | value  |
| <b>shum</b>   | Specific humidity (g/kg)  | value  |
| <b>rhum</b>   | Relative humidity (%)   | value  |
| <b>wetblb</b> | Wetbulb temperature (K)   | value  |
| <b>thetaw</b> | Wetbulb potential temperature (K)   | value  |
| <b>thetae</b> | Equivalent potential temperature (K)  | value  |
| <b>thetav</b> | Virtual potential temperature (K)   | value  |
| <b>theta</b>  | Potential temperature (K)   | value  |
| <b>vtemp</b>  | Virtual temperature (K)   | value  |
| <b>fr</b>     | Flight rules (number and string: 0=VFR, 1=MVFR, 2=IFR)                        | value=0, string=VFR<br>value=1, string=MVFR<br>value=2, string=IFR |
| <b>front</b>  | Fronts (special plot type)  |  |

These parameters can be accessed by name if they have not been redefined in the **sfccalc.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear. Also, time can be specified for use with precipitation (**anal-6acc::prec::%.2f** with give 6 hour precipitation).

There are several derived variables from this list which are defined in the **sfccalc.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable      | Description                                    | Plots  |
|---------------|--|--------|
| <b>wind</b>   | A vector quantity of <uwndg,vwndg>             | vector |
| <b>vect</b>   | Wind vector plot                               | vector |
| <b>strm</b>   | Streamline plot                                | vector |
| <b>pres</b>   | Sea level pressure (mb)                        | value  |
| <b>spres</b>  | Station pressure (mb)                          | value  |
| <b>ptend</b>  | Pressure tendency (mb)                         | value  |
| <b>prec6</b>  | 6 hour precipitation (in) to 2 decimal places  | value  |
| <b>prec12</b> | 12 hour precipitation (in) to 2 decimal places | value  |
| <b>prec24</b> | 24 hour precipitation (in) to 2 decimal places | value  |
| <b>maxt</b>   | 24 hour maximum temperature (F)                | value  |
| <b>mint</b>   | 24 hour minimum temperature (F)                | value  |
| <b>maxt6</b>  | 6 hour maximum temperature (F)                 | value  |
| <b>mint6</b>  | 6 hour minimum temperature (F)                 | value  |
| <b>conv</b>   | Wind convergence (/s) positive is convergence  | grid   |
| <b>mconv</b>  | Moisture convergence (/s)                      | grid   |
| <b>rvort</b>  | Relative vorticity (/s)                        | grid   |
| <b>avort</b>  | Absolute vorticity (/s)                        | grid   |
| <b>tadv</b>   | Temperature advection (K/s)                    | grid   |
| <b>madv</b>   | Moisture advection (/s)                        | grid   |
| <b>vadv</b>   | Vorticity advection (/s^2)                     | grid   |

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the converted file is opened and data read in. Any additional computations are performed to produce a derived variable and fit the data to a grid. The background map will plot first unless the output is a color fill contour in which the map plots last. The gridding process uses an objective analysis scheme to produce the grids. The parameters for this process are specified in the **object\_param** resource. Once the grids have been created, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.
- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

## EXAMPLES

```
sfccalc -cu=la -re=mw -va=temp -pl=cf -in=2 -de=d
```

This produces a color fill contour plot of temperature for the midwest region at an interval of 2.

```
sfccalc -cu=la -re=mw -va=temp -of=wxp -pl=none
```

This will generate grid file based on the **sfc\_grd** name convention. A specific filename can be used instead of **wxp**.

## FILES

- **sao.cty** - the surface station location database file
- **sfccalc.var** - the variable menu and definitions

## SEE ALSO

- [sacvt](#) - the surface data decoding program
- [smcvt](#) - the synoptic data decoding program.
- [sfcwx](#) - the surface data plotting program
- [statlog](#) - the surface meteogram plotting program

# SFCWX

## NAME

**sfcwx** - Surface data plotting program

## SYNOPSIS

**sfcwx** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description   |
|------------------------------|----------------------------|-------------------|---|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.   |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxddef           | Sets the name of the resource file.   |
| <b>-na=name</b>              | <a href="#">name</a>       | sfcwx             | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode   |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>gridding information - <b>out3</b></li> </ul>  |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.   |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files which includes <b>front</b> files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.  |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.   |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.   |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.   |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | sfc_cvt           | Specifies the input file name tag. The default is <b>sfc_cvt</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well. In order to plot fronts, specify <b>front</b>  |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of raw or grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII grid or raw file</li> <li><b>wxpb</b> - WXP binary grid file</li> <li><b>raw</b> - WXP ASCII raw file</li> <li><b>cdf</b> - NetCDF binary grid file</li> </ul> Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>sfc_raw</b> or <b>sfc_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type: |

|   |                              |  |   |
|---|------------------------------|--|---|
|   |                              |  | wxp:sfc_grd<br><br>A full filename can be specified which can contain name convention tags.   |
| <b>-pf=</b> <i>file_param</i>   | <a href="#">file_param</a>   | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul> |
| <b>-cf=</b> <i>cityfile</i>   | <a href="#">city_file</a>    | <b>sao.cty</b>   | Specifies the name of the city database file used in finding station locations.   |
| <b>-mf=</b> <i>mapfile</i> [,...]   | <a href="#">map_file</a>     | <b>wxp.map</b> or <b>cont.dat</b> , <b>country.dat</b> for domains outside North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]   | <a href="#">current</a>      | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=</b> <i>hour</i>   | <a href="#">hour</a>         | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh=</b> <i>num_hour</i>   | <a href="#">num_hour</a>     | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.   |
| <b>-id=</b> <i>identifier</i>   | <a href="#">identifier</a>   | None   | Used to plot a specific observation. If not specified, all stations will be plotted.  |
| <b>-pd=</b> <i>domain</i>   | <a href="#">plot_domain</a>  | User prompt<br>Batch: first domain in <b>wxp.reg</b> file.                               | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-gd=</b> <i>grid_domain</i>  | <a href="#">grid_domain</a>  | Same as plot domain  | This specifies the domain to use when fitting data to a grid. If this is not specified, it is the <b>plot_domain</b> .  |
| <b>-re=</b> <i>region</i>   | <a href="#">region</a>       | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-pr=</b> <i>prior</i>  | <a href="#">stat_prior</a>   | Depends on domain size   | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.  |
| <b>-va=</b> <i>variable</i>   | <a href="#">variable</a>     | User prompt<br>Batch: first variable listed in the <b>sfcwx.var</b> file                 | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.  |
| <b>-oa=</b> <i>smooth</i><br>[, <i>radinf</i> ][, <i>passes</i> ]<br>[, <i>converg</i> ]<br>[, <i>min_sta</i> ] | <a href="#">object_param</a> | <b>1,3,1,,2,1</b>  | Specifies the objective analysis parameters. There are 5 parameters including smoothing, radius of influence, number of passes, convergence parameter and minimum number of stations.   |
| <b>-pl=</b> <i>plot_type</i>  | <a href="#">plot_type</a>    | <b>Data Plot</b><br>User prompt if plotting grids  | Specifies the type of output plot. By default, this is a simple data plot. To produce a contour or vector plot, use the following plot types:   |

|                                   |                              |   |  |
|-----------------------------------|------------------------------|---|--|
|                                   |                              | Batch: fill contours or vectors                           | <p><b>Contours:</b></p> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <p><b>Vectors:</b></p> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>           | <a href="#">plot_scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm=format</b>                 | <a href="#">plot_format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp=param</b><br>[,param...]   | <a href="#">plot_param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.  |
| <b>-dr=drawable[,...]</b>         | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.   |
| <b>-la=label[,...]</b>            | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-in=interval</b>               | <a href="#">con_interval</a> | None for plots,<br>User prompt for grids. Batch: <b>0</b> | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.   |
| <b>-cb=base</b>                   | <a href="#">con_base</a>     | Lowest contour value, maximum wind speed                  | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.   |
| <b>-fl=font_list</b>              | <a href="#">font_list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.   |
| <b>-ct=color_table</b>            | <a href="#">color_table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |
| <b>-cot=color[:attr]</b>          | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse plots.           | Specifies the text color for labels above and below the plot.  |
| <b>-cod=color[:attr]</b>          | <a href="#">color_data</a>   | <b>green</b>  | Specifies the color for plotted data.  |
| <b>-com=color[:attr]</b>          | <a href="#">color_map</a>    | <b>red</b>  | Specifies the color for background maps.   |
| <b>-coln=color[:attr]</b>         | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>                                     | Specifies the color for lat/lon lines.   |
| <b>-cowl=color[:attr]</b>         | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse plots.           | Specifies the color for wind barbs.  |
| <b>-codd=color[:attr]</b>         | <a href="#">color_cloud</a>  | <b>white</b><br><b>black</b> for inverse plots.           | Specifies the color for cloud cover symbols.   |
| <b>-cowx=color[:attr]</b>         | <a href="#">color_wx</a>     | <b>yellow</b><br><b>brown</b> for inverse plots           | Specifies the color for weather symbols  |
| <b>-cofr=color[:attr],</b><br>... | <a href="#">color_front</a>  | A predefined color set                                    | Specifies the colors for fronts and pressure systems.  |
| <b>-cocm=color[:attr]</b>         | <a href="#">color_cmap</a>   | <b>black</b>  | Specifies the color for color fill contour maps.   |
| <b>-coco=color[:attr]</b>         | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse                  | Specifies the color for line contours.   |

|   |                              |  |  |
|---|------------------------------|--|--|
|   |                              | plots.   |  |
| <b>-cof</b> =[val:]color<br>[:attr],... | <a href="#">color_fill</a>   | A predefined color set based on the <b>wxp.clr</b> color table | Specifies the color fill contour color set. See the <b>color_fill</b> resource for more details.   |
| <b>-cola</b> =color[:attr]              | <a href="#">color_label</a>  | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the line contour label color.  |
| <b>-cocl</b> =color[:attr]              | <a href="#">color_clabel</a> | <b>black</b>   | Specifies the color fill contour label color.  |
| <b>-de</b> =device[,...]                | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                           | Specifies the output device.   |
| <b>-ti</b> =title                       | <a href="#">title</a>        | Program name   | Specifies the window title.  |
| <b>-ic</b>                              | <a href="#">icon</a>         | <b>Off</b>   | Specifies whether to open window in iconified mode.  |
| <b>-bg</b> =color                       | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots                 | Specifies the background color of the window.  |
| <b>-ge</b> =geometry                    | <a href="#">geometry</a>     | <b>640x480</b>   | Specifies the window and paper geometry.   |
| filename[#seq]<br>[filename[#seq]...]   | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: current= <b>la</b>               | The name of the surface converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified to generate a loop. |

## DESCRIPTION

This program plots surface data on a background map. The input to the program is a surface converted file produced by either the [sacvt](#) or [smcvt](#) programs. Then, based on a plot domain and a variable selected by the user, a plot is generated. Overlay plots, contour plots and loops can also be generated if the appropriate options are set.

The programs starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **sfc\_cvt** tag but this can be changed with the [in\\_file](#) resource. To generate a loop, the [num\\_hour](#) resource can be specified. This will tell the program to loop through this number of hours.

Next, the program prompts the user for the plot domain. This is done through a region menu which has predefined regions available. If needed a full plot domain can be specified to get a precise domain. The domain can be interactively changed using the mouse once the plot has been generated.

After the region has been set, the city database is read in and a cross-reference table is setup. Since converted data is not saved with location information, the program will refer to this table to location each station in the input file. The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the **sfcwx.var** file. The internal variables are:

### Internal Variables

| Variable     | Description and units          | Plots   |
|--------------|--------------------------------|---|
| <b>city</b>  | Station ID                     | string  |
| <b>temp</b>  | Temperature (F)                | value   |
| <b>ttd</b>   | Temperature and dewpoint (F)   | value <sub>0</sub> =temp,<br>value <sub>1</sub> =dewp |
| <b>dewp</b>  | Dewpoint (F)                   | value   |
| <b>dewd</b>  | Dewpoint depression (F)        | value   |
| <b>wdir</b>  | Wind direction (deg)           | value   |
| <b>wspd</b>  | Wind speed (knt)               | value   |
| <b>wgst</b>  | Wind gust (knt)                | value   |
| <b>wbrbc</b> | Wind speed and direction (knt) | value <sub>0</sub> =dir,                              |

|               |  |  |
|---------------|--|--|
| <b>wbrb</b>   |  | value <sub>1</sub> =spd  |
| <b>wind</b>   |  |  |
| <b>uwnd</b>   | U wind component - EW relative (m/s)   | value  |
| <b>vwnd</b>   | V wind component - NS relative (m/s)   | value  |
| <b>uwndg</b>  | U wind component - grid relative (m/s)   | value  |
| <b>vwndg</b>  | V wind component - grid relative (m/s)   | value  |
| <b>alt</b>    | Altimeter setting (in_Hg)  | value  |
| <b>pres</b>   | Sea level pressure (mb) use level parameter to get station/surface pressure  | value  |
| <b>pralt</b>  | Sea level pressure or altimeter setting if sea level pressure is missing (mb)  | value  |
| <b>elev</b>   | Station elevation (m)  | value  |
| <b>wx</b>     | Present weather (SAO string)   | string   |
| <b>vis</b>    | Visibility (mi)  | value, formatted string  |
| <b>clldcl</b> | Cloud ceiling (100 ft), string is "" for clear skies, value is 500 for clear   | value, formatted string  |
| <b>clldcl</b> | Cloud ceiling code (from synoptic code 0-9)  | value  |
| <b>cldcv</b>  | Cloud cover (cloud string: <b>C,F,S,O,X,0-8</b> ), converted to % for contouring.  | value=%, string  |
| <b>ptend</b>  | Pressure tendency (mb)   | value, string=trend code   |
| <b>prec</b>   | Precipitation (in) use time parameter to specify which type (3,6,12,24 hour)   | value  |
| <b>extt</b>   | Extreme temperature (F) uses maximum if available, else minimum.<br>Use time parameter to specify which type (6,24 hour) | value  |
| <b>maxt</b>   | Maximum temperature (F) use time parameter to specify which type (6,24 hour)   | value  |
| <b>mint</b>   | Minimum temperature (F) use time parameter to specify which type (6,24 hour)   | value  |
| <b>snwdp</b>  | Snow depth (in)  | value  |
| <b>nlcd</b>   | Amount of low clouds (code string)   | value=%, string=code   |
| <b>lcl</b>    | Low cloud type (code string)   | string   |
| <b>mcl</b>    | Middle cloud type (code string)  | string   |
| <b>hcl</b>    | High cloud type (code string)  | string   |
| <b>solar</b>  | Equivalent solar radiation (min)   | value  |
| <b>sst</b>    | Sea surface temperature (C)  | value  |
| <b>wavper</b> | Wave period (sec)  | value  |
| <b>wavhgt</b> | Wave height (m)  | value  |
| <b>vpres</b>  | Vapor pressure (mb)  | value  |
| <b>svpres</b> | Saturation vapor pressure (mb)   | value  |
| <b>wchilt</b> | Wind chill temperature (K)   | value  |
| <b>wchilf</b> | Wind chill factor (W/m <sup>2</sup> )  | value  |
| <b>heat</b>   | Heat index (K)   | value  |
| <b>humit</b>  | Humiture (F)   | value  |
| <b>mrat</b>   | Mixing ratio (g/kg)  | value  |
| <b>smrat</b>  | Saturation mixing ration (g/kg)  | value  |
| <b>shum</b>   | Specific humidity (g/kg)   | value  |
| <b>rhum</b>   | Relative humidity (%)  | value  |
| <b>wetblb</b> | Wetbulb temperature (K)  | value  |
| <b>thetaw</b> | Wetbulb potential temperature (K)  | value  |
| <b>thetae</b> | Equivalent potential temperature (K)   | value  |
| <b>thetav</b> | Virtual potential temperature (K)  | value  |
| <b>theta</b>  | Potential temperature (K)  | value  |
| <b>vtemp</b>  | Virtual temperature (K)  | value  |
| <b>fr</b>     | Flight rules (number and string: )   | value=0, string=VFR<br>value=1, string=MVFR<br>value=2, string=IFR |
| <b>front</b>  | Fronts (special plot type)   |  |

These parameters can be accessed by name if they have not been redefined in the **sfcwx.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear. Also, time can be specified for use with precipitation (**anal-6acc::prec::%.2f** with give 6 hour precipitation).

There are several derived variables from this list which are defined in the **sfcwx.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable      | Description   | Plots  |
|---------------|---|--------|
| <b>all</b>    | All data, a composite plot of temperature, dewpoint, sea level pressure, cloud cover, winds and present weather.  | comp   |
| <b>allc</b>   | Same as <b>all</b> except temperatures in Celsius.  | comp   |
| <b>full</b>   | All data, a more complete station model plot of station ID, temperature, dewpoint, sea level pressure, cloud cover, winds, present weather, visibility, pressure tendency (value+symbol), cloud ceiling (code), low, medium, high cloud symbols | comp   |
| <b>fullc</b>  | Same as <b>all</b> except temperatures in Celsius.  | comp   |
| <b>depict</b> | Weather depiction plot including flight rules (contoured), cloud cover, cloud ceiling and present weather.  | comp   |
| <b>wind</b>   | A vector quantity of <uwndg,vwndg>  | vector |
| <b>vect</b>   | Wind vector plot  | vector |
| <b>strm</b>   | Streamline plot   | vector |
| <b>pres</b>   | Sea level pressure (mb) to 1 decimal place  | value  |
| <b>spres</b>  | Station pressure (mb) to 1 decimal place  | value  |
| <b>alt</b>    | Altimeter setting (in_Hg) to 2 decimal places   | value  |
| <b>lmhcd</b>  | Composite plot of low, medium and high cloud types as symbols   | comp   |
| <b>ptend</b>  | Pressure tendency (mb)  | value  |
| <b>ptype</b>  | Pressure tendency trend as a symbol   | symbol |
| <b>prec6</b>  | 6 hour precipitation (in) to 2 decimal places   | value  |
| <b>prec12</b> | 12 hour precipitation (in) to 2 decimal places  | value  |
| <b>prec24</b> | 24 hour precipitation (in) to 2 decimal places  | value  |
| <b>maxt</b>   | 24 hour maximum temperature (F)   | value  |
| <b>mint</b>   | 24 hour minimum temperature (F)   | value  |
| <b>maxt6</b>  | 6 hour maximum temperature (F)  | value  |
| <b>mint6</b>  | 6 hour minimum temperature (F)  | value  |
| <b>fr</b>     | Flight rules plotted as text (VFR, MVFR, IFR)   | text   |
| <b>mrat</b>   | Mixing ratio (g/kg) to 1 decimal place  | value  |
| <b>shum</b>   | Specific humidity (g/kg) to 1 decimal place   | value  |
| <b>conv</b>   | Wind convergence (/s) positive is convergence   | grid   |
| <b>mconv</b>  | Moisture convergence (/s)   | grid   |
| <b>rvort</b>  | Relative vorticity (/s)   | grid   |
| <b>avort</b>  | Absolute vorticity (/s)   | grid   |
| <b>tadv</b>   | Temperature advection (K/s)   | grid   |
| <b>madv</b>   | Moisture advection (/s)   | grid   |
| <b>vadv</b>   | Vorticity advection (/s^2)  | grid   |

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the converted file is opened and data read in. Any additional computations are performed and the data are plotted. The background map will plot first unless the output is a color fill contour in which the map plots last. If a simple plot is specified such as "**temp**", the data will be plotted to 0 decimal places on the map. Formatting of the number can be done with the **plot\_format** resource. Composite plots can either be specified with the **variable** resource (.var file, resource file, command line or variable prompt). The **variable** resource allows the user to control much of the data formatting.

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

### Contour/Vector Plots

By default, this program plots data but it can produce grids that can either be contoured or plotted as vectors or streamlines. By specifying the **plot\_type** resource to a contour or vector type, the program will generate gridded fields rather than plotting the data. The gridding process uses an objective analysis scheme to produce the grids. The parameters for this process are specified in the **object\_param** resource. Once the grids have been created, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display in much the same fashion as for the data plot.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.
- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

### Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

### Front Data

The sfcwx program will plot coded front data in the ASUS1 products. The **in\_file** resource must be set to **front** (sets **frt\_dat** file name tag) for **sfcwx** to use front files instead of surface converted data. High and Low pressure centers along with cold, warm, stationary and occluded fronts and troughs are plotted. The colors are controlled by the **color\_front** resource.

### EXAMPLES

```
sfcwx -cu=la -re=mw -va=all -de=d
```

This plots a surface composite plot containing temperature, dewpoint, pressure, clouds, winds and weather for the midwest region.

```
sfcwx -cu=la -re=mw -va=temp -of=raw -pl=none
```

This will generate raw file based on the **sfc\_raw** name convention. A specific filename can be used instead of **raw**.

```
sfcwx -cu=la -re=mw -va=temp -pl=cf -in=2 -de=d
```

This will produce a contour plot of temperature. Without the "-pl=cf", the program will just plot the temperature data.

```
sfcwx -cu=la -if=front -re=us
```

This plots front files.

## FILES

- **sao.cty** - the surface station location database file
- **sfcwx.var** - variable menu and definitions

## SEE ALSO

- [sacvt](#) - the surface data decoding program
- [smcvt](#) - the synoptic data decoding program.
- [sfccalc](#) - the surface data contouring program
- [statlog](#) - the surface meteogram plotting program

# SHEFCVT

## NAME

**shefcvt** - SHEF data conversion program

## SYNOPSIS

**shefcvt** [*parameters...*] *filename*

## PARAMETERS

| Command Line                             | Resource                   | Default  | Description  |
|--|----------------------------|--|--|
| <b>-h</b>                                | <a href="#">help</a>       | No   | Lists basic help information.  |
| <b>-df=filename</b>                      | <a href="#">default</a>    | .wxpdef  | Sets the name of the resource file.  |
| <b>-na=name</b>                          | <a href="#">name</a>       | shefcvt  | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                               | <a href="#">batch</a>      | No   | Run program in batch mode  |
| <b>-me=level</b>                         | <a href="#">message</a>    | out2   | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>decoded output -<b>out3</b></li> </ul>  |
| <b>-fp=filepath</b>                      | <a href="#">file_path</a>  | current directory                              | Specifies location of database files.  |
| <b>-dp=datapath</b>                      | <a href="#">data_path</a>  | current directory                              | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>                       | <a href="#">raw_path</a>   | current directory                              | Specifies the location (path) of the output raw data files. This may be modified in the name convention file.  |
| <b>-nc=name_conv</b>                     | <a href="#">name_conv</a>  | name_conv                                      | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>                       | <a href="#">in_file</a>    | shef_dat                                       | Specifies the input file name tag. The default is <b>shef_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-of=out_file[,out...]</b>             | <a href="#">out_file</a>   | shef_raw                                       | The <b>out_file</b> resource specifies the output file name tag. The default is <b>shef_raw</b> but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-pf=file_param</b>                    | <a href="#">file_param</a> | User prompt<br>Batch: <b>use</b>               | This specifies how to treat files that exist: <ul style="list-style-type: none"> <li><b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li><b>Over</b> - overwrite the existing file</li> <li><b>App</b> - append the information onto the end of the existing file</li> <li><b>Y</b> - same as overwrite</li> <li><b>N</b> - same as use.</li> </ul> |
| <b>-cu=[hour]la</b>                      | <a href="#">current</a>    | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=hour</b>                          | <a href="#">hour</a>       | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-pa=param</b><br>[ <i>,param...</i> ] | <a href="#">parameter</a>  | None   | Extra parameters: <ul style="list-style-type: none"> <li><b>date:date</b> - puts this date in output raw file</li> </ul>   |
| <i>filename</i> [ <i>#seq</i> ]          | <a href="#">filename</a>   | None<br>User Pompt<br>Batch: <b>current=la</b> | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-WXP files.   |

## DESCRIPTION

The SHEF decoding program will search for and decode SHEF reports from SRUS products. The decoder takes SHEF data which is essentially tagged numbers and converts it to a raw file format which is compatible the WXP. A sample of SHEF data are:

```

** SRUS23 KTOP 272358 ***
SRUS23 KTOP 272358
FSD
.A HLTK1 1227 C DH18/TA 17/TX 36/TN 17

** SRUS51 KWBC 280001 ***
SRUS51 KWBC 280001
^NMCRRATX3
: && HADS TPR REPORT TPR3 FOR LBB
.B NMC 1228 DH0001/PCIRG/PPKRG/HGIRG/HPIRG
EART2 DD272330/16.59/ .00//
KNBT2 DD272130/12.00/ .00/2.85/
NLKT2 DD272130/9.74/ .00//
TKLT2 DD272330/10.33/ .00/4.23/
BILT2 DD272130/7.81/ .00//
STST2 DD272130/12.93/ .00//
BCPT2 DD272130/12.41/ .00//
MMCT2 DD272121/10.14/ .00/ .01/
TKST2 DD272130/16.03/ .00/6.66/
TBRT2 DD272130/12.81/ .00//1928.99
.END

** SRUS53 KWBC 280001 ***
SRUS53 KWBC 280001
^NMCRRAMN
: && HADS TPR REPORT TPR1 FOR MSP
.B NMC 1228 DH0001/HGIRG/PPKRG/TAIRG/HPIRG/HTIRG/PCIRG
ATKM5 DD2721/7.60/////
RKF5 DD2722/5.30/////
ANKM5 DD2723/7.64/////
SSPM5 DD2722/87.01/////
GTBW3 DD2723/4.75/////
SCFW3 DD2722/2.18/////

```

These data are decoded into a raw file with a name based on the **shef\_raw** name convention. The output raw file from the above data would be:

```

WXPRAW
  0Z 28 DEC 93
SHEF Data
HLTK1 TA:17 TX:36 TN:17
EART2 PCIRG:16.59 PPKRG:.00
KNBT2 PCIRG:12.00 PPKRG:.00 HGIRG:2.85
NLKT2 PCIRG:9.74 PPKRG:.00
TKLT2 PCIRG:10.33 PPKRG:.00 HGIRG:4.23
BILT2 PCIRG:7.81 PPKRG:.00
STST2 PCIRG:12.93 PPKRG:.00
BCPT2 PCIRG:12.41 PPKRG:.00
MMCT2 PCIRG:10.14 PPKRG:.00 HGIRG:.01

```

```

TKST2 PCIRG:16.03 PPKRG:.00 HGIRG:6.66
TBRT2 PCIRG:12.81 PPKRG:.00 HPIRG:1928.99
ATKM5 HGIRG:7.60
RKFM5 HGIRG:5.30
ANKM5 HGIRG:7.64
SSPM5 HGIRG:87.01
GTBW3 HGIRG:4.75
SCFW3 HGIRG:2.18

```

As can be seen, the column headings on the **.B** reports and data descriptors on the **.A** reports become prefixes on the data saved to the raw file. Individual data types can be extracted by using either **mapplt** or **rawmath**.

If the message level is set to **out3**, some decoded information is printed to the screen:

```

SHEFA .A HLTk1 1227 C DH18/TA 17/TX 36/TN 17
ID: HLTk1
  DATE: 1227
  ZONE: C
  TIME: H 18
  TA-Temperature: 17 F
  TX-Maximum temperature: 36 F
  TN-Minimum temperature: 17 F
SHEFB .B NMC 1228 DH0001/PCIRG/PPKRG/HGIRG/HPIRG
ID: NMC
  DATE: 1228
  TIME: H 0001
  TYPE: PCIRG
  TYPE: PPKRG
  TYPE: HGIRG
  TYPE: HPIRG
ID: EART2
  TIME: D 272330
  PCIRG-Precipitation accumulated: 16.59 in
  PARAM: PPKRG .00
ID: KNBT2
  TIME: D 272130
  PCIRG-Precipitation accumulated: 12.00 in
  PARAM: PPKRG .00
  HGIRG-River stage: 2.85 ft
ID: NLKT2
  TIME: D 272130
  PCIRG-Precipitation accumulated: 9.74 in
  PARAM: PPKRG .00
ID: TKLT2
  TIME: D 272330
  PCIRG-Precipitation accumulated: 10.33 in
  PARAM: PPKRG .00
  HGIRG-River stage: 4.23 ft

```

## EXAMPLES

```
shefcvt -cu=la
```

will convert the latest SHEF file and create a raw file.

To plot SHEF data, mapplt is set up so that it can pick a variable and display it.

```
mapplt -if=12112296.shef-HGIRG -re=40,-86,.3 -cf=shef.cty \
```

```
-mf=state.map,ind_cnty.map
```

Also rawmath can extract a field from the raw file.

```
rawmath ext 12112296.shef-HGIRG = - -fm=5.2
```

Would yield a file such as:

```
WXPRAW
12Z 22 NOV 96
SHEF Data-HGIRG
KNBT2  2.72
TKLT2  6.17
MMCT2  0.01
TKST2  6.61
ATKM5  7.17
RKFM5  4.73
ANKM5  4.38
...
```

Rawmath has the capability of creating time series of SHEF data. For example, rawmath could be used to create a new raw file with:

```
rawmath series 12111996.shef-HGIRG:NOV19 12112096.shef-HGIRG:NOV20 \
12112196.shef-HGIRG:NOV21 12112296.shef-HGIRG:NOV22 = - -id=LAFI3 \
-fm=5.2 "SHEF-HGIRG for LAFI3"
```

Would yield:

```
WXPRAW
12Z 19 NOV 93
SHEF-HGIRG
NOV19 15.43
NOV20 13.75
NOV21 11.05
NOV22  9.39
```

The mapplt program will create a time-value plot:

```
rawmath series 12111996.shef-HGIRG:NOV19 \
12112096.shef-HGIRG:NOV20 \
12112196.shef-HGIRG:NOV21 \
12112296.shef-HGIRG:NOV22 = - \
-id=LAFI3 -fm=4.2 "SHEF-HGIRG for LAFI3" | \
mapplt -pd=cat,0,5,2,2,3,15 -if=std \
-pp=line,mark:1,lalo:1:1 -de=d
```

This uses the series function to extract **HGIRG** data from the SHEF raw files. This is labeled by day for the station **LAFI3**.

## FILES

- **shef.cty** - the SHEF station location database file

## SEE ALSO

- [mapplt](#) - plots raw files
- [rawmath](#) - the raw file math program

# SMCVT

## NAME

**smcvt** - Synoptic surface data conversion program

## SYNOPSIS

**smcvt** [*parameters...*] *filename*

## PARAMETERS

| Command Line         | Resource                  | Default           | Description  |
|----------------------|---------------------------|-------------------|--|
| <b>-h</b>            | <a href="#">help</a>      | No                | Lists basic help information.  |
| <b>-df=filename</b>  | <a href="#">default</a>   | .wxpdef           | Sets the name of the resource file.  |
| <b>-na=name</b>      | <a href="#">name</a>      | smcvt             | Specifies the name used in resource file parsing.  |
| <b>-ba</b>           | <a href="#">batch</a>     | No                | Run program in batch mode  |
| <b>-me=level</b>     | <a href="#">message</a>   | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• decoded output - <b>out3a</b></li> <li>• raw data and format - <b>out3c</b></li> <li>• not in database - <b>out3c</b></li> </ul>  |
| <b>-fp=filepath</b>  | <a href="#">file_path</a> | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>  | <a href="#">data_path</a> | current directory | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>   | <a href="#">con_path</a>  | current directory | Specifies the location (path) of the output converted data files. This may be modified in the name convention file.  |
| <b>-nc=name_conv</b> | <a href="#">name_conv</a> | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>   | <a href="#">in_file</a>   | syn_dat           | Specifies the input file name tag. The default is <b>syn_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-ou=output</b>    | <a href="#">output</a>    | wxp, all          | Specifies what type of output file is to be created: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII converted file. Name convention set to <b>syn_cvt_wxp</b>.</li> <li>• <b>cdf</b> - NetCDF binary converted file. Name convention set to <b>syn_cvt_cdf</b>.</li> </ul> Specifies which type of synoptic data to save: <ul style="list-style-type: none"> <li>• <b>all</b> - all synoptic reports</li> <li>• <b>land</b> - all land based reports</li> <li>• <b>cman</b> - CMAN station</li> <li>• <b>buoy</b> - stationary buoy</li> <li>• <b>ship</b> - ship report</li> <li>• <b>dribu</b> - drifting buoy</li> </ul> |
| <b>-of=out_file</b>  | <a href="#">out_file</a>  | syn_cvt           | The <b>out_file</b> resource specifies the output file name tag. The default is <b>syn_cvt</b> but it can be modified to any value in the filename convention file. A full name convention can be specified as well.<br><br>Also, can specifies what type of output file is to be created: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII converted file</li> <li>• <b>cdf</b> - NetCDF binary converted file</li> </ul>  |

|  |                             |   |  |
|--|-----------------------------|---|--|
| <b>-pf=</b> <i>file_param</i>  | <a href="#">file_param</a>  | User prompt<br>Batch: <b>use</b>                | This specifies how to treat files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul>   |
| <b>-cf=</b> <i>cityfile</i>  | <a href="#">city_file</a>   | None  | Specifies the name of the city database file used in finding station locations.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]                                  | <a href="#">current</a>     | None  | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=</b> <i>hour</i>  | <a href="#">hour</a>        | None  | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=</b> <i>num_hour</i>  | <a href="#">num_hour</a>    | <b>0</b>  | This specifies the number of hours that will be searched for hourly data.  |
| <b>-dh=</b> <i>decode_hour</i>   | <a href="#">decode_hour</a> | None  | Specifies a specific decode time when using more than one file.  |
| <b>-id=</b> identifier   | <a href="#">identifier</a>  | None  | Used to decode a specific observation. If not specified, all stations will be decoded.   |
| <b>-pa=</b> <i>param</i><br>[, <i>param</i> ...]                         | <a href="#">parameter</a>   | None  | Extra parameters: <ul style="list-style-type: none"> <li>• <b>wmo</b> - save station with its WMO ID (5 digit).</li> <li>• <b>id</b> - save station with its ICAO ID (4 letter).</li> <li>• <b>id_enc</b> - save station with its ICAO ID and encode one if missing.</li> <li>• <b>pln</b> - print single line report</li> <li>• <b>plns</b> - print the 2 line report</li> <li>• <b>psyn</b> - print using synoptic print format.</li> <li>• <b>pfull</b> - print using the full multiline report</li> <li>• <b>range:pre:post</b> - specifies time range for valid data.(default: -90:90)</li> <li>• <b>all</b> - save all data to output file.</li> <li>• <b>dup</b> - remove duplicate reports prior to saving to file</li> <li>• <b>update</b> - remove duplicates and update existing reports with corrections.</li> </ul> |
| <i>filename</i> [# <i>seq</i> ]<br>[ <i>filename</i> [# <i>seq</i> ]...] | <a href="#">filename</a>    | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified.  |

## DESCRIPTION

The intent of the surface data conversion process is to combine all the surface data (SAO, METAR, Synopt, Buoy, Ship, DRIBU, CMAN) into one format so that a single program can plot all the data. The surface conversion process is split up into two programs **sacvt** (METAR/SAO) and **smcvt** (Synoptic, Buoy, Ship, CMAN, DRIBU). **smcvt** program handles hourly data in each of the above listed formats. All stations for a particular report time are decoded and put into an output converted data file.

The **smcvt** program will first prompt the user for the input raw data file. This file should contain the synoptic reports for a given time. It is recommended that these files be split up on 3 or 6 hourly boundaries since this type of data is reported once every 6 hours with intermediate reports every 3 hours. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **syn\_dat** tag but this can be changed with the [in\\_file](#) resource. If the data stretches over an hour, the [num\\_hour](#) resource can be specified to search over multiple files.

Once the input file is determined, the program determines the decode time from the filename specified or that of the **current** resource. If decode time is different from that of the specified file, the [decode hour](#) resource must be specified. This resource can also be used to set decode time range limits which is useful if decoding data from off times such as :20 after the hour. Once the decode time is set, the output converted file is opened for output. The name convention is determined from the **syn\_cvt** file convention tag. If the converted file exists, the user will be prompted whether to overwrite, append or use the existing file. The append option leaves the current data in the converted file and adds the new station data onto the end of the file. If the use option is specified, the converted file is not modified and the surface data conversion program exits.

The data are parsed for individual station reports which are then passed through the synoptic decode module. Once decoded, the output is saved to the output converted file in the order they were decoded. This is not true if the **dup** or **update** options are specified. The remove duplicate (**dup**) option removes duplicate reports from the output converted file. The update station (**update**) option not only removes duplicate stations but updates station data with corrections and amendments.

The conversion process is silent. By default, none of the output is displayed to the screen as an attempt to speed up the conversion process. Decoded output can be displayed if the **out3a** message level is set:

```

  ID  Cht Cv  Vis  PWx  SLPres  T  Td  WD-WS  WG  Te  PT  Rn6 Rn24  SC  L:M:H
Processing file: /rainbow/data/ddplus/00060797.syn
72405  30 8  15.0      1020.7  63  50 120- 7   67  0.3      8:7:-
72597   1 28.1      1011.0  80  47 300- 6   80 -1.5     1:0:0
72652   1 15.0      1013.1  82  49  0- 0   82 -1.0     0:0:2
72594   0 10.0      1013.4  63  53 360-12  65 -0.2
91165   1 15.0      1018.7  82  73  70-15   75 -1.2     0.02  1:0:0
78650  45 8  1.2  TRW- 1006.4  80  77 270- 4   87  1.6  0.47  9:2:-
78655  45 8  15.6      73  68 100- 2   80      8:4:0
78663  15 8  2.5      1006.5  71  70  0- 0   72 -1.1  0.08  0.09  8:2:0
91285  15 7  6.9  RW- 1017.2  79  72  40- 6   72 -1.4     0.20  8:0:0
91182  30 3  15.0      1018.0  87  67  90-16  75 -1.2     0.00  8:0:0
b51003      1015.6  80      40-10
b51002      1015.9  76      100-17

```

If the raw information is to be displayed, use the **out3** message level:

```

  ID  Cht Cv  Vis  PWx  SLPres  T  Td  WD-WS  WG  Te  PT  Rn6 Rn24  SC  L:M:H
Processing file: /rainbow/data/ddplus/00060797.syn
1:72405:72405 32574 81207 10173 20102 30183 40207 53003 8287/ 333 10195 20118 555
90700
72405  30 8  15.0      1020.7  63  50 120- 7   67  0.3      8:7:-
1:72597:72597 32783 13006 10266 20085 39639 40110 57015 81100 333 10266 20078 555
90700
72597   1 28.1      1011.0  80  47 300- 6   80 -1.5     1:0:0
1:72652:72652 32874 10000 10276 20097 39605 40131 58010 80002 333 10278 20128 81057
72652   1 15.0      1013.1  82  49  0- 0   82 -1.0     0:0:2
1:72594:72594 32966 03612 10170 20116 30134 40134 58002 333 10181 20132
72594   0 10.0      1013.4  63  53 360-12  65 -0.2
1:91165:91165 12474 10715 10277 20230 30135 40187 58012 69954 81100 333 20241 561//
59002 70005 555 90700
91165   1 15.0      1018.7  82  73  70-15   75 -1.2     0.02  1:0:0
1:78650:78650 21620 82704 10264 20250 40064 52016 79598 8792/ 222// 20603 333 072/
10304 20264 32/// 58016 60121 85827 82930 84640
78650  45 8  1.2  TRW- 1006.4  80  77 270- 4   87  1.6  0.47  9:2:-
1:78655:78655 32675 81002 10226 20200 3//// 4//// 5//// 70200 83840 333 0173/ 10264
20206 30/// 5//// 83637 86370
78655  45 8  15.6      73  68 100- 2   80      8:4:0
1:78663:78663 21440 80000 10218 20212 39384 40065 57011 70386 88820 333 021// 10222
20215 32/// 60021 70023 83810 87633 85465
78663  15 8  2.5      1006.5  71  70  0- 0   72 -1.1  0.08  0.09  8:2:0

```

```

1:91285:91285 11461 70406 10262 20225 30159 40172 58014 60054 78082 87800 333 20222
562// 59005 70046 555 90700
91285 15 7 6.9 RW- 1017.2 79 72 40- 6 72 -1.4 0.20 8:0:0
1:91182:91182 12574 30916 10308 20196 30174 40180 57012 69904 83800 333 20239 56999
58000 555 90700
91182 30 3 15.0 1018.0 87 67 90-16 75 -1.2 0.00 8:0:0
1:72613:72613 32289 43409 10067 20009 38132 54000 81108 333 10077 20000
72613 4 4 46.9 44 34 340- 9 46 0.0 1:0:8
4:51003:51003 07001 99191 71608 46/// /0405 10264 40156 92349 22200 00270 10803
70017 333 91206 555 11055 22058
b51003 1015.6 80 40-10
4:51002:51002 07001 99172 71578 46/// /1009 10242 40159 92349 22200 00272 11304
70020 333 91211 555 11093 22098
b51002 1015.9 76 100-17

```

The raw report is preceded by the report type (1=land, 2=ship, 4=buoy, 8=dribu, 16=cman). For more information on output formats, see the [text output section of the users guide](#).

There are two types of output data file types available. The ASCII wxp Flat file data type saves all available information in a columnar format which may be edited with a standard text editor. The netCDF data format is a binary format which may be produced by specifying **cdf** for the output file type.

The output file may save the ID either as the WMO number (as broadcasted) or as a ICAO ID (3 or 4 letter). If the station does not have an ICAO ID, one will be derived from the WMO number using the 2 letter country prefix and the last 3 numbers of the WMO number (**-pa=id\_enc**). Country prefixes are listed in the [Global Station Information Appendix](#).

Station IDs for buoys are always preceded with a "b". Station IDs for ships are always preceded with a "s". DRIBUs don't have an ID and are listed as "DRIBU".

**NOTE:** **smcvt** must be run before any surface plotting and gridding programs may be run.

**NOTE:** **smcvt** produces the same converted file name as does **sacvt**. If the same hour is converted by **sacvt**, the program will prompt the user for overwrite or append.

## EXAMPLES

```

smcvt -cu=$current -dh=cu-15+15 -of=sfc_cvt -pa=update -pf=app \
-me=print -ou=ship,cman,buoy,dribu

```

This converts the synoptic data minus land stations. The decode hour is set to make sure only those reports within 15 minutes of the report time are saved. The output goes to the same file as the **sacvt** program (with the METAR/SAO reports). The converted data is appended to the end of the METAR data.

```

smcvt -cu=$current -dh=cu-15+15 -if=cman_dat -pa=update \
-of=sfc_cvt -pf=app -me=print

```

This produces a decode file from CMAN data which is stored in a different file from the regular synoptic data as pointed to by the **cman\_dat** tag. The data are appended to the same surface converted file as **sacvt** and only those reports within 15 minutes of the report time are saved. The update parameter tells the program to only saved updated reports.

## FILES

- **syn.cty** - the surface station location database file

## SEE ALSO

- [sacvt](#) - the synoptic data decoding program
- [sfcwx](#) - the surface data plotting program

- [sfccalc](#) - the surface data contouring program
- [statlog](#) - the surface meteogram plotting program

# STATLOG

## NAME

**statlog** - Surface meteogram plotting program

## SYNOPSIS

**statlog** [*parameters...*] *filename*

## PARAMETERS

| Command Line                       | Resource                   | Default  | Description  |
|------------------------------------|----------------------------|--|--|
| <b>-h</b>                          | <a href="#">help</a>       | No   | Lists basic help information.  |
| <b>-df=<i>filename</i></b>         | <a href="#">default</a>    | .wxpdef  | Sets the name of the resource file.  |
| <b>-na=<i>name</i></b>             | <a href="#">name</a>       | statlog  | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                         | <a href="#">batch</a>      | No   | Run program in batch mode  |
| <b>-me=<i>level</i></b>            | <a href="#">message</a>    | out2   | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>   |
| <b>-fp=<i>filepath</i></b>         | <a href="#">file_path</a>  | current directory                                | Specifies location of database files.  |
| <b>-dp=<i>datapath</i></b>         | <a href="#">data_path</a>  | current directory                                | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=<i>conpath</i></b>          | <a href="#">con_path</a>   | current directory                                | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=<i>rawpath</i></b>          | <a href="#">raw_path</a>   | current directory                                | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=<i>gridpath</i></b>         | <a href="#">grid_path</a>  | current directory                                | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=<i>imagepath</i></b>        | <a href="#">image_path</a> | current directory                                | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=<i>name_conv</i></b>        | <a href="#">name_conv</a>  | name_conv  | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=<i>in_file</i></b>          | <a href="#">in_file</a>    | sfc_cvt (for sfc data)<br>mos_dat (for MOS data) | Specifies the input file name tag. The default is <b>sfc_cvt</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-mo=<i>model</i></b>            | <a href="#">model</a>      | sfc  | The type of data to use in the plot: <ul style="list-style-type: none"> <li><b>sfc</b> - use surface data</li> <li><b>ngm</b> - use NGM MOS data</li> </ul>  |
| <b>-cu=[<i>hour</i>]<b>la</b>]</b> | <a href="#">current</a>    | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=<i>hour</i></b>             | <a href="#">hour</a>       | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=<i>num_hour</i></b>         | <a href="#">num_hour</a>   | -12  | This specifies the number of hours that will be used in the plot. If this is not specified, 12 hours of data will plot. This is ignored for MOS data.  |
| <b>-id=<i>identifier</i></b>       | <a href="#">identifier</a> | None<br>User prompt                              | Specifies the station to use in plotting data.   |
| <b>-va=<i>variable</i></b>         | <a href="#">variable</a>   | User prompt<br>Batch: first                      | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by   |

|                                 |                             |   |   |
|---------------------------------|-----------------------------|---|---|
|                                 |                             | variable listed in the <b>statlog.var</b> file  | commas for overlay plots. A list of valid variables is given below.   |
| <b>-sc=scale_factor</b>         | <a href="#">plot_scale</a>  | <b>1.0</b>                                      | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>               | <a href="#">plot_format</a> | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.                      |
| <b>-pp=param</b><br>[,param...] | <a href="#">plot_param</a>  | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>       | <a href="#">draw</a>        | <b>all</b>                                      | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>          | <a href="#">label</a>       | Program defaults                                | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels. |
| <b>-fl=font_list</b>            | <a href="#">font_list</a>   | <b>modern.fnt</b>                               | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>          | <a href="#">color_table</a> | <b>wxp.clr</b>                                  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>        | <a href="#">color_text</a>  | <b>white</b><br><b>black</b> for inverse plots. | Specifies the text color for labels above and below the plot.   |
| <b>-cod=color[:attr]</b>        | <a href="#">color_data</a>  | <b>green</b>                                    | Specifies the color for plotted data.   |
| <b>-cogr=color[:attr]</b>       | <a href="#">color_grid</a>  | <b>red</b>                                      | Specifies the color for background grids.   |
| <b>-cows=color[:attr]</b>       | <a href="#">color_wind</a>  | <b>white</b><br><b>black</b> for inverse plots. | Specifies the color for wind barbs.   |
| <b>-cocd=color[:attr]</b>       | <a href="#">color_cloud</a> | <b>white</b><br><b>black</b> for inverse plots. | Specifies the color for cloud cover symbols.  |
| <b>-cowx=color[:attr]</b>       | <a href="#">color_wx</a>    | <b>yellow</b><br><b>brown</b> for inverse plots | Specifies the color for weather symbols   |
| <b>-de=device[,...]</b>         | <a href="#">device</a>      | User prompt<br>Batch: <b>display</b>            | Specifies the output device.  |
| <b>-ti=title</b>                | <a href="#">title</a>       | Program name                                    | Specifies the window title.   |
| <b>-ic</b>                      | <a href="#">icon</a>        | <b>Off</b>                                      | Specifies whether to open window in iconified mode.   |
| <b>-bg=color</b>                | <a href="#">background</a>  | <b>black</b><br><b>white</b> for inverse plots  | Specifies the background color of the window.   |
| <b>-ge=geometry</b>             | <a href="#">geometry</a>    | <b>640x480</b>                                  | Specifies the window and paper geometry.  |
| <b>filename[#seq]</b>           | <a href="#">filename</a>    | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the surface converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files.                                    |

## DESCRIPTION

This program plots a time series of surface data on a meteogram. The input to the program is either a surface converted file produced by either the [sacvt](#) or [smcvt](#) programs or NGM MOS data. Then, based on a variable selected by the user, a plot is generated.

### Surface Data Meteogram

The surface data meteogram will plot a time series of data. The number of hours is based on the [num\\_hour](#) resource. The default is 12 hours but often 24 hours is used.

The programs starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the [sfc\\_cvt](#) tag but this can be changed with the [in\\_file](#) resource.

Next, the program prompts the user for the station identifier to use in the plot.

The program will now prompt the user for the particular output variable to plot. Even though a meteogram is considered to be a set of plots, there is the ability to plot individual variables versus time. The program will show a menu of commonly used variables but many more are defined internally by the program and in the [statlog.var](#) file. The internal variables are:

#### Internal Variables

| Variable                                   | Description and units   | Plots   |
|--|---|---|
| <b>temp</b>                                | Temperature (F)   | value   |
| <b>dewp</b>                                | Dewpoint (F)  | value   |
| <b>dewd</b>                                | Dewpoint depression (F)   | value   |
| <b>wdir</b>                                | Wind direction (deg)  | value   |
| <b>wspd</b>                                | Wind speed (knt)  | value   |
| <b>wgst</b>                                | Wind gust (knt) (not in MOS)  | value   |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)  | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)  | value   |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)  | value   |
| <b>uwndg</b>                               | U wind component - grid relative (m/s)  | value   |
| <b>vwndg</b>                               | V wind component - grid relative (m/s)  | value   |
| <b>alt</b>                                 | Altimeter setting (in_Hg) (not in MOS)  | value   |
| <b>pres</b>                                | Sea level pressure (mb) use level parameter to get station/surface pressure   | value   |
| <b>pralt</b>                               | Sea level pressure or altimeter setting if sea level pressure is missing (mb)   | value   |
| <b>elev</b>                                | Station elevation (m)   | value   |
| <b>wx</b>                                  | Present weather (SAO string)  | string  |
| <b>vis</b>                                 | Visibility (mi)   | value, formatted string                             |
| <b>cldc1</b>                               | Cloud ceiling (100 ft)  | value, formatted string                             |
| <b>cldc1c</b>                              | Cloud ceiling code (from synoptic code 0-9)   | value   |
| <b>cldcv</b>                               | Cloud cover (cloud string)  | value=%, string                                     |
| <b>ptend</b>                               | Pressure tendency (mb, trend is a string) (not in MOS)  | value, string=trend code                            |
| <b>prec</b>                                | Precipitation (in) use time parameter to specify which type (3,6,12,24 hour)  | value   |
| <b>extt</b>                                | Extreme temperature (F) uses maximum if available, else minimum. Use time parameter to specify which type (6,24 hour) | value   |
| <b>maxt</b>                                | Maximum temperature (F) use time parameter to specify which type (6,24 hour)  | value   |
| <b>mint</b>                                | Minimum temperature (F) use time parameter to specify which type (6,24 hour)  | value   |
| <b>snwdp</b>                               | Snow depth (in)   | value   |
| <b>nlcld</b>                               | Amount of low clouds (code string)  | value=%, string=code                                |
| <b>lclcd</b>                               | Low cloud type (not in MOS)   | string  |
| <b>mclcd</b>                               | Middle cloud type (not in MOS)  | string  |
| <b>hclcd</b>                               | High cloud type (not in MOS)  | string  |
| <b>solar</b>                               | Equivalent solar radiation (min) (not in MOS)   | value   |
| <b>sst</b>                                 | Sea surface temperature (C) (not in MOS)  | value   |
| <b>wavper</b>                              | Wave period (sec) (not in MOS)  | value   |
| <b>wavhgt</b>                              | Wave height (m) (not in MOS)  | value   |

|               |  |  |
|---------------|--|--|
| <b>vpres</b>  | Vapor pressure (mb) (not in MOS)                       | value  |
| <b>svpres</b> | Saturation vapor pressure (mb) (not in MOS)            | value  |
| <b>wchilt</b> | Wind chill temperature (K)                             | value  |
| <b>wchilf</b> | Wind chill factor (W/m <sup>2</sup> )                  | value  |
| <b>heat</b>   | Heat index (K)   | value  |
| <b>humit</b>  | Humiture (F)   | value  |
| <b>mrat</b>   | Mixing ratio (g/kg)                                    | value  |
| <b>smrat</b>  | Saturation mixing ration (g/kg)                        | value  |
| <b>shum</b>   | Specific humidity (g/kg)                               | value  |
| <b>rhum</b>   | Relative humidity (%)                                  | value  |
| <b>wetblb</b> | Wetbulb temperature (K)                                | value  |
| <b>thetaw</b> | Wetbulb potential temperature (K)                      | value  |
| <b>thetae</b> | Equivalent potential temperature (K)                   | value  |
| <b>thetav</b> | Virtual potential temperature (K)                      | value  |
| <b>theta</b>  | Potential temperature (K)                              | value  |
| <b>vtemp</b>  | Virtual temperature (K)                                | value  |
| <b>fr</b>     | Flight rules (number and string: 0=VFR, 1=MVFR, 2=IFR) | value=0, string=VFR<br>value=1, string=MVFR<br>value=2, string=IFR |

These parameters can be accessed by name if they have not been redefined in the **statlog.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

There are several derived variables from this list which are defined in the **statlog.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable    | Description  | Plots |
|-------------|--|-------|
| <b>all</b>  | All data, a composite plot of temperature, dewpoint, sea level pressure, cloud cover, winds and present weather. | comp  |
| <b>allc</b> | Same as <b>all</b> except temperatures in Celsius.   | comp  |

Once the variable is selected, the program will open and search through the set of hourly files to get all the data for the time series plot. A table of information for the station over the requested hours will print:

Data for KIND from 1200Z 3 OCT 97 to 0000Z 4 OCT 97

| ID   | TIME | T  | TD | RH | DIR | SPD | GST | ALT | SLP | VIS | CIL | COV | WX | MAX | MIN | PR6 | PR24 | SC |
|------|------|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|-----|------|----|
| KIND | 1156 | 55 | 51 | 87 | 180 | 6   |     | 993 | 133 | 10  | 180 | SCT |    | 58  | 55  |     |      |    |
| KIND | 1256 | 59 | 53 | 81 | 200 | 8   |     | 993 | 134 | 10  | 210 | SCT |    |     |     |     |      |    |
| KIND | 1356 | 64 | 54 | 70 | 220 | 13  |     | 994 | 135 | 10  | 230 | FEW |    |     |     |     |      |    |
| KIND | 1456 | 69 | 55 | 61 | 210 | 12  |     | 993 | 131 | 10  | 230 | FEW |    |     |     |     |      |    |
| KIND | 1556 | 73 | 56 | 55 | 210 | 13  |     | 991 | 125 | 10  | 230 | FEW |    |     |     |     |      |    |
| KIND | 1656 | 76 | 57 | 52 | 220 | 11  |     | 991 | 123 | 10  | 230 | FEW |    |     |     |     |      |    |
| KIND | 1756 | 78 | 58 | 50 | 210 | 10  |     | 989 | 118 | 10  |     | CLR |    | 79  | 55  |     |      |    |
| KIND | 1856 | 81 | 59 | 47 | 210 | 11  | 21  | 987 | 109 | 10  |     | CLR |    |     |     |     |      |    |
| KIND | 1956 | 82 | 60 | 48 | 210 | 14  | 20  | 985 | 104 | 10  |     | CLR |    |     |     |     |      |    |
| KIND | 2056 | 82 | 60 | 48 | 210 | 14  |     | 985 | 101 | 10  |     | CLR |    |     |     |     |      |    |
| KIND | 2156 | 81 | 60 | 49 | 220 | 13  |     | 985 | 102 | 10  | 250 | FEW |    |     |     |     |      |    |
| KIND | 2256 | 77 | 60 | 56 | 220 | 8   |     | 986 | 106 | 10  | 85  | FEW |    |     |     |     |      |    |
| KIND | 2356 | 74 | 61 | 64 | 180 | 8   |     | 987 | 111 | 10  | 85  | FEW |    | 82  | 74  |     |      |    |

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the meteogram will be plotted. For the all data plot, this is a composite of various plots:

- **TEMP - temperature and dewpoint chart:** This shows temperature as the top line and dewpoint as the bottom line.
- **EXTT - extreme temperature:** The maximum or minimum temperatures are plotted below the chart at specific reporting times. 12Z and 18Z are generally low temperatures and 0Z and 6Z are high temperatures.
- **WX - present weather data:** This shows the present weather plotted as a weather symbol.
- **SNOW - snow cover:** This shows reported snow cover in inches.
- **PREC - precipitation:** This shows precipitation totals in inches. The values are 24 hour totals generally at 12Z, 6 hour totals at 0Z, 6Z and 18Z, 3 hour totals elsewhere.
- **VIS - visibility:** This shows the horizontal visibility in miles.
- **WGST - wind gusts:** This reports wind gusts in knots if reported.
- **WIND - winds and cloud cover:** This plots the standard wind barb and cloud cover symbol. Up is north.
- **CLD - Cloud chart:** This chart gives cloud layer information. The vertical axis is a logarithmic scale of the cloud base height in feet ranging from the 100 to 50,000 feet. Each cloud layer base is plotted as horizontal lines. Clear skies are plotted as a 'C'. Scattered cloud layers (1/8th to 3/8th coverage) are plotted as a single short dash "-". Broken cloud layers (4/8th to 7/8th coverage) are plotted as two short dashes "- -". Overcast layers are plotted as a single long dash "---". The actual cloud ceiling is displayed below the chart in 100s of feet if the ceiling is below 10,000 feet.
- **PRALT - Pressure chart:** This chart plots sea level pressure (or altimeter setting if pressure not reported) in millibars.
- **Time:** This is the report hour in universal time. The actual time range is displayed above the plot.

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## Meteogram Plotting

The default for each variable selected is a time series plot/chart. The chart has a range from just above the maximum to just below the minimum. The Y axis is labeled with the variable name and the units. The label is rotated counterclockwise. If the plot type is not **plot**, the data are printed (type=**value**) or plotted in the case of weather symbols (type=**wx**), cloud cover (type=**cloud**) or wind barbs (type=**cbarb**). If the variable is **cldcl**, the cloud ceilings are plotted using a logarithmic scale of the cloud base height in feet ranging from the 100 to 50,000 feet. Each cloud layer base is plotted as horizontal lines. Clear skies are plotted as a 'C'. Scattered cloud layers (1/8th to 3/8th coverage) are plotted as a single short dash "-". Broken cloud layers (4/8th to 7/8th coverage) are plotted as two short dashes "- -". Overcast layers are plotted as a single long dash "---".

If more than one variable is listed, the program stacks up the time series plots from top to bottom. If the variable is a group {**var1,var2**}, the program plots those on top of each other or on the same chart. The size of these plots can be set with the "**sz=size**" attribute. The size is the fraction of the largest side of the window. For most plots, the maximum size is about **.8**. Here is an example of the composite plot listed above:

```
all      All_Data          1 temp {temp,dewp} [F :plot:sz=.18],\
                        extt      [F :value],\
                        wx         [ :wx],\
                        snwdp     [in :value],\
                        prec       [in :value:%.2f],\
                        vis        [mi :value],\
                        wgst      [knt:value],\
                        wind { \
                          wind     [knt:cbarb],\
                          cldcv    [% :cloud] },\
                        cld        [ft :plot:sz=.14],\
                        cldcl      [100_ft:data],\
                        pres       [mb :plot:sz=.14]
```

## MOS Data Meteogram

The MOS data meteogram will plot a time series of forecast data. The NGM MOS data lists forecast data up to 60 hours. In this mode, a meteogram of forecast data will be produced. To trigger this mode, the **model** resource must be set to "ngm".

The programs starts off by prompting the user for input MOS file name. The user may specify the input file either via the command line or through the **current** resource. The input naming convention is specified by the **mos\_dat** tag but this can be changed with the **in\_file** resource.

Next, the program prompts the user for the station identifier to use in the plot. Remember that MOS data is stored by the 3 letter identifier in most cases.

The program will now prompt the user for the particular output variable to plot. Since the MOS data is close to regular surface data, most of the above variables are available. The program attempts to get sea level pressure data from the NGM-UA data. If that is available, most parameters can be computed. The program now lists the forecast data in a table:

```

NGM MOS data for IND from 1800Z  3 OCT 97 to 0000Z  6 OCT 97

ID  TIME  T  TD  RH  DIR  SPD  GST  ALT  SLP  VIS  CIL  COV  WX  MAX  MIN  PR6  PR24  SC
IND 1800 76 60 58 220 15      120 10 250 SCT
IND 2100 78 60 54 220 13      115 10 250 SCT
IND 0000 72 61 68 210  7      110 10 250 SCT
IND 0300 67 61 81 200  6      115 10    CLR
IND 0600 65 60 84 190  7      120 10    CLR
IND 0900 63 60 90 220  9      130 10 250 SCT
IND 1200 63 60 90 230 10      140 10 250 BKN          61
IND 1500 72 62 71 230 12      140 10 250 BKN
IND 1800 78 62 58 240 15      140 10 250 BKN
IND 2100 79 61 54 240 13      145 10 250 BKN
IND 0000 73 60 64 230  8      150 10 250 SCT          82
IND 0300 66 58 75 240  5      155      CLR
IND 0600 61 56 84 250  4      160 10    CLR
IND 0900 59 55 87 210  4      170      CLR
IND 1200 58 55 90 200  3      180 10    CLR          56
IND 1500 69 58 68 210  8      180 10    CLR
IND 1800 77 57 50 220 11      180 10    CLR
IND 2100 78 57 48 220 10      180 10    CLR
IND 0000 71 57 61 200  6      180 10    SCT          80

```

The program prompts the user for the output device and then produces the MOS meteogram.

## EXAMPLES

```
statlog -cu=la -id=KIND -va=all -de=d
```

This plots a composite 12 hour meteogram for KIND.

```
statlog -cu=la -id=KIND -nh=-24 -va=all -de=d
```

This creates a 24 hour meteogram.

```
statlog -mo=ngm -cu=la -id=IND -va=all -de=d
```

This plots a 60 hour forecast meteogram for IND.

```
statlog -cu=la -id=KPHL -va=Temp{temp,dewp},wspd -de=d
```

This plots temperature, dewpoint on the same chart and wind speed on a different chart.

## FILES

- **sao.cty** - the surface station location database file
- **statlog.var** - variable menu and definitions

## SEE ALSO

- [sacvt](#) - the surface data decoding program
- [smcvt](#) - the synoptic data decoding program.

# STDATMS

## NAME

**stdatms** - Standard atmosphere computation program

## SYNOPSIS

**stdatms** [*parameters*] [*value*]

## PARAMETERS

| Command Line         | Resource                  | Default             | Description  |
|----------------------|---------------------------|---------------------|--|
| <b>-h</b>            | <a href="#">help</a>      | No                  | Lists basic help information.  |
| <b>-df=filename</b>  | <a href="#">default</a>   | .wxddef             | Sets the name of the resource file.  |
| <b>-na=name</b>      | <a href="#">name</a>      | stdatms             | Specifies the name used in resource file parsing.  |
| <b>-ba</b>           | <a href="#">batch</a>     | No                  | Run program in batch mode  |
| <b>-me=level</b>     | <a href="#">message</a>   | out2                | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>output values - <b>print</b></li> </ul> |
| <b>-va=units</b>     | <a href="#">variable</a>  | none<br>User prompt | Specifies which input variable and its units:<br><b>Hm, Hft, PmmHg, PinHg, Pmb, Ppsi, Ppas, Patm</b>                       |
| <b>-pa=parameter</b> | <a href="#">parameter</a> | none                | Specifies additional parameters.   |
| <i>value</i>         | <a href="#">value</a>     | none<br>User prompt | Specifies the input value.   |

## DESCRIPTION

The standard atmosphere calculation program relates pressure, height and temperature. The user is prompted for a height or pressure in one of the following units:

| Variable/Type | Explanation                        |
|---------------|------------------------------------|
| <b>Hm</b>     | Height in meters                   |
| <b>Hft</b>    | Height in feet                     |
| <b>PmmHg</b>  | Pressure in mm mercury             |
| <b>PinHg</b>  | Pressure in inches mercury         |
| <b>Pmb</b>    | Pressure in millibars              |
| <b>Ppsi</b>   | Pressure in pounds per square inch |
| <b>Ppas</b>   | Pressure in Pascals                |
| <b>Patm</b>   | Pressure in atmospheres            |

Valid ranges include: -100 meters < height < 20000 meters, 1050 mb > pressure > 55 mb. Then, you enter one of the above parameters and units through a menu listing.

The program then calculates the height, pressure and temperature (in various units) at that specific level for a standard atmosphere. The output appears as:

```

Standard Atmosphere Data
Height      :          3000.0000 meters
              9842.5197 feet
Pressure    :          701.0852 mb
              70108.5156 pascals
              525.8822 mm (Hg)
              20.7032 in (Hg)
              10.1684 PSI
              0.6919 atmospheres

```

Temperature : -4.5000 Celsius  
23.9000 Fahrenheit

**FILES**

**SEE ALSO**

# SUNCALC

## NAME

**suncalc** - Computes sunrise/sunset times.

## SYNOPSIS

**suncalc** [*parameters...*] *location date*

## PARAMETERS

| Command Line                    | Resource                  | Default             | Description  |
|---------------------------------|---------------------------|---------------------|--|
| <b>-h</b>                       | <a href="#">help</a>      | No                  | Lists basic help information.  |
| <b>-df=filename</b>             | <a href="#">default</a>   | .wxpdef             | Sets the name of the resource file.  |
| <b>-na=name</b>                 | <a href="#">name</a>      | tafcvt              | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                      | <a href="#">batch</a>     | No                  | Run program in batch mode  |
| <b>-me=level</b>                | <a href="#">message</a>   | out2                | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>output - <b>print</b></li> </ul>  |
| <b>-fp=filepath</b>             | <a href="#">file_path</a> | current directory   | Specifies location of database files.  |
| <b>-cf=cityfile</b>             | <a href="#">city_file</a> | None                | Specifies the name of the city database file used in finding station names.  |
| <b>-pa=param</b><br>[,param...] | <a href="#">parameter</a> | None                | Extra parameters (none currently)  |
| <i>location date</i>            | <a href="#">value</a>     | None<br>User Prompt | The location is either: <i>lat,lon[,zone]</i> or <i>ident[,zone]</i><br><br>The date is either: <i>yyyy/mm/dd/dd[+nn]</i> or <b>cyear</b> , <b>cmonth</b> , <b>cday[+nn]</b> |

## DESCRIPTION

This program does general sunrise and sunset computations. The input to the program is two parameters:

- **location** -- This is specified with either the latitude and longitude of the site or a station identifier which is looked up in the **sa0.cty** city database file for location. Remember that latitude is positive north and longitude is positive east. The time zone is computed from the longitude but this often does not give the true time zone. The zone parameter is an hour offset from GMT. For example, EST is "-5". There is an optional "N" parameter if the site does not go on daylight time.
- **date** -- This is specified either with the exact date or with the current date which can be **cyear**, **cmonth**, or **cday**. An optional minute increment can be specified "+nn".

The "DST" environment variable can be set to toggle the zone computation into daylight time.

```
setenv DST yes
```

The program output varies on the input to the program.

- **year only** -- This specifies to give sunrise, sunset information for an entire year.

```
Lat: 40.42 Lon: -86.93 Zone: -5.0
Year: 1997
Day Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1 0811-1732 0757-1806 0721-1839 0631-1912 0547-1943 0520-2012 0521-2022 0545-2002 0615-1919 0644-1830 0718-1744 0752-1722
2 0811-1733 0756-1807 0720-1841 0630-1913 0546-1944 0519-2013 0522-2022 0546-2001 0616-1918 0645-1828 0719-1743 0753-1722
3 0811-1734 0755-1809 0718-1842 0628-1915 0544-1945 0519-2013 0522-2021 0547-2000 0617-1916 0646-1826 0720-1742 0754-1721
4 0811-1735 0754-1810 0717-1843 0626-1916 0543-1946 0518-2014 0523-2021 0548-1959 0618-1915 0647-1825 0721-1741 0755-1721
...
24 0804-1757 0729-1834 0644-1904 0556-1936 0524-2006 0518-2022 0538-2010 0607-1932 0637-1841 0709-1755 0744-1724 0809-1726
25 0803-1758 0727-1835 0643-1905 0555-1937 0523-2006 0519-2022 0539-2009 0608-1930 0638-1840 0710-1753 0745-1724 0809-1727
26 0802-1759 0726-1836 0641-1906 0553-1938 0523-2007 0519-2022 0540-2008 0609-1929 0639-1838 0711-1752 0746-1723 0810-1727
```

```

27 0801-1800 0724-1837 0639-1907 0552-1939 0522-2008 0520-2022 0541-2007 0610-1927 0640-1836 0712-1751 0748-1723 0810-1728
28 0801-1801 0723-1838 0638-1908 0551-1940 0522-2009 0520-2022 0541-2006 0611-1926 0641-1835 0713-1749 0749-1723 0810-1729
29 0800-1803 0721-1839 0636-1909 0549-1941 0521-2010 0520-2022 0542-2005 0612-1924 0642-1833 0714-1748 0750-1722 0810-1730
30 0759-1804      0635-1910 0548-1942 0521-2010 0521-2022 0543-2004 0613-1923 0643-1831 0715-1747 0751-1722 0811-1730
31 0758-1805      0633-1911      0520-2011      0544-2003 0614-1921      0717-1746      0811-1731
    
```

- **year and month only** -- This specifies to give more detailed information on sunrise, sunset, twilight and solar noon for each day during the month.

Lat: 40.42 Lon: -86.93 Zone: -5.0  
 Year: 1997 Month: 6

| Day | Dawn     | Sunrise  | Sunset   | Dusk     | = | Noon     | DayLen   | Twilight |
|-----|----------|----------|----------|----------|---|----------|----------|----------|
| 1   | 04:47:11 | 05:19:37 | 20:11:52 | 20:44:23 | = | 12:45:45 | 14:52:15 | 00:32:26 |
| 2   | 04:46:42 | 05:19:12 | 20:12:35 | 20:45:09 | = | 12:45:54 | 14:53:22 | 00:32:30 |
| 3   | 04:46:15 | 05:18:50 | 20:13:16 | 20:45:55 | = | 12:46:03 | 14:54:26 | 00:32:34 |
| 4   | 04:45:51 | 05:18:29 | 20:13:56 | 20:46:38 | = | 12:46:13 | 14:55:26 | 00:32:38 |
| 5   | 04:45:29 | 05:18:11 | 20:14:35 | 20:47:21 | = | 12:46:23 | 14:56:23 | 00:32:42 |
| ... |          |          |          |          |   |          |          |          |
| 27  | 04:46:21 | 05:19:30 | 20:21:59 | 20:55:06 | = | 12:50:44 | 15:02:29 | 00:33:09 |
| 28  | 04:46:47 | 05:19:54 | 20:21:58 | 20:55:04 | = | 12:50:56 | 15:02:03 | 00:33:07 |
| 29  | 04:47:14 | 05:20:20 | 20:21:55 | 20:54:59 | = | 12:51:08 | 15:01:35 | 00:33:05 |
| 30  | 04:47:44 | 05:20:48 | 20:21:50 | 20:54:52 | = | 12:51:19 | 15:01:02 | 00:33:03 |

- **year, month and day** -- This specifies to give sunrise, sunset, extended twilight information for that day

Lat: 40.42 Lon: -86.93 Zone: -5.0  
 Year: 1997 Month: 6 Day: 4

```

Ast Dawn: 03:16:51
Nau Dawn: 04:04:36
Dawn: 04:45:51
Sunrise: 05:18:29
Sunset: 20:13:56
Dusk: 20:46:38
Nau Dusk: 21:28:00
Ast Dusk: 22:15:58
-----
Sol Noon: 12:46:13
Daylength: 14:55:26
Twilight: 00:32:38
    
```

- **year, month, day and minute increment** -- This specifies to give detailed solar location information including elevation, azimuth, zenith angle and declination. The minute increment sets the difference between each computation which is 10 minutes in the sample output.

Lat: 40.42 Lon: -86.93 Zone: -5.0  
 Year: 1997 Month: 6 Day: 4

| Time                | UTC      | Elev   | Elev(R) | Azimuth | Zenith | Declin | RtAscen |
|---------------------|----------|--------|---------|---------|--------|--------|---------|
| 00:00:00            | 05:00:00 | -26.24 |         | 348.15  | 116.24 | 22.43  | 72.24   |
| 00:10:00            | 05:10:00 | -26.59 |         | 350.70  | 116.59 | 22.43  | 72.25   |
| 00:20:00            | 05:20:00 | -26.85 |         | 353.27  | 116.85 | 22.44  | 72.26   |
| 00:30:00            | 05:30:00 | -27.03 |         | 355.85  | 117.03 | 22.44  | 72.27   |
| 00:40:00            | 05:40:00 | -27.13 |         | 358.45  | 117.13 | 22.44  | 72.27   |
| 00:50:00            | 05:50:00 | -27.14 |         | 1.04    | 117.14 | 22.44  | 72.28   |
| ...                 |          |        |         |         |        |        |         |
| 03:00:00            | 08:00:00 | -19.83 |         | 32.84   | 109.83 | 22.45  | 72.37   |
| 03:10:00            | 08:10:00 | -18.76 |         | 35.01   | 108.76 | 22.45  | 72.38   |
| ---- AST DAWN ----- |          |        |         |         |        |        |         |
| 03:20:00            | 08:20:00 | -17.64 |         | 37.14   | 107.64 | 22.45  | 72.39   |
| 03:30:00            | 08:30:00 | -16.46 |         | 39.22   | 106.46 | 22.45  | 72.39   |
| 03:40:00            | 08:40:00 | -15.23 |         | 41.25   | 105.23 | 22.45  | 72.40   |
| 03:50:00            | 08:50:00 | -13.95 |         | 43.24   | 103.95 | 22.45  | 72.41   |
| 04:00:00            | 09:00:00 | -12.63 |         | 45.18   | 102.63 | 22.45  | 72.42   |
| ---- NAU DAWN ----- |          |        |         |         |        |        |         |

|                     |          |        |      |       |        |       |       |
|---------------------|----------|--------|------|-------|--------|-------|-------|
| 04:10:00            | 09:10:00 | -11.25 |      | 47.08 | 101.25 | 22.45 | 72.42 |
| 04:20:00            | 09:20:00 | -9.84  |      | 48.94 | 99.84  | 22.45 | 72.43 |
| 04:30:00            | 09:30:00 | -8.38  |      | 50.75 | 98.38  | 22.46 | 72.44 |
| 04:40:00            | 09:40:00 | -6.89  |      | 52.53 | 96.89  | 22.46 | 72.44 |
| ----- DAWN -----    |          |        |      |       |        |       |       |
| 04:50:00            | 09:50:00 | -5.36  |      | 54.27 | 95.36  | 22.46 | 72.45 |
| 05:00:00            | 10:00:00 | -3.80  |      | 55.98 | 93.80  | 22.46 | 72.46 |
| 05:10:00            | 10:10:00 | -2.21  |      | 57.66 | 92.21  | 22.46 | 72.47 |
| ----- SUNRISE ----- |          |        |      |       |        |       |       |
| 05:20:00            | 10:20:00 | -0.58  |      | 59.30 | 90.58  | 22.46 | 72.47 |
| 05:30:00            | 10:30:00 | 1.07   | 1.42 | 60.92 | 88.93  | 22.46 | 72.48 |
| 05:40:00            | 10:40:00 | 2.74   | 2.98 | 62.51 | 87.26  | 22.46 | 72.49 |
| 05:50:00            | 10:50:00 | 4.44   | 4.62 | 64.07 | 85.56  | 22.46 | 72.49 |
| 06:00:00            | 11:00:00 | 6.17   | 6.30 | 65.62 | 83.83  | 22.46 | 72.50 |
| 06:10:00            | 11:10:00 | 7.91   | 8.02 | 67.15 | 82.09  | 22.46 | 72.51 |
| 06:20:00            | 11:20:00 | 9.67   | 9.76 | 68.66 | 80.33  | 22.46 | 72.52 |

The output has the following data:

- **sunrise** - the time at which the first part of the sun appears above the horizon in the morning.
- **sunset** - the time at this the last part of the sun disappears below the horizon in the evening.
- **dawn** or **civil dawn** - this is the time at which the sun is 6 degrees below the horizon in the morning. Civil dawn is defined as that time at which there is enough light for objects to be distinguishable and that outdoor activities can commence.
- **nautical dawn** - this is the time at which the sun is 12 degrees below the horizon in the morning. Nautical dawn is defined as that time at which there is just enough sunlight for objects to be distinguishable.
- **astronomical dawn** - this is the time at which the sun is 18 degrees below the horizon in the morning. Astronomical dawn is that point in time at which the sun starts lightening the sky. Prior to this time, the sky is completely dark
- **dusk** or **civil dusk** - this is the time at which the sun is 6 degrees below the horizon in the evening. At this time objects are distinguishable but there is no longer enough light to perform any outdoor activities.
- **nautical dusk** - this is the time at which the sun is 12 degrees below the horizon in the evening. At this time, objects are no longer distinguishable.
- **astronomical dusk** - this is the time at which the sun is 18 degrees below the horizon in the evening. At this time the sun no longer illuminates the sky.
- **solar noon** - this is the time at which the sun is the highest in the sky. This time varies through the year due to the change in speed of the earth's orbit around the sun.
- **daylength** - the total time the sun is above the horizon
- **twilight** - the average time of civil twilight which is the time between civil dawn and sunrise and sunset and civil dusk.
- **elevation** - the angle that the sun is above the horizon (assuming no atmosphere)
- **elevation (r)** - the angle that the sun appears to be above the horizon assuming the refraction of the earth's atmosphere.
- **azimuth** - the direction that the sun appears to be over with 0 degrees being north, 90 being east, 180 being south and 270 being west.
- **zenith angle** - the angle between the sun and directly overhead, the zenith
- **declination** - the latitude that the sun is directly over which is ~23N in the summer, ~23S in the winter and 0 at the equinoxes.
- **right ascension** - the celestial longitude of the sun. This value is 0 at the vernal equinox, 90 at the summer solstice, 180 at the autumnal equinox and 270 at the winter solstice.

## FILES

- **sao.cty** - used to cross reference station identifiers with location
- **cities.raw** - a list of stations with location and time zone data.

## SEE ALSO

# TAFCVT

## NAME

**tafcvt** - Terminal forecast (TAF format) decoder

## SYNOPSIS

**tafcvt** [*parameters...*] *filename*

## PARAMETERS

| Command Line                             | Resource                   | Default            | Description  |
|--|----------------------------|--------------------|--|
| <b>-h</b>                                | <a href="#">help</a>       | No                 | Lists basic help information.  |
| <b>-df=filename</b>                      | <a href="#">default</a>    | .wpxdef            | Sets the name of the resource file.  |
| <b>-na=name</b>                          | <a href="#">name</a>       | tafcvt             | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                               | <a href="#">batch</a>      | No                 | Run program in batch mode  |
| <b>-me=level</b>                         | <a href="#">message</a>    | out2               | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• selected decoded output - <b>print</b></li> <li>• unselected decoded output - <b>out3</b></li> <li>• additional information - <b>out4</b></li> </ul>  |
| <b>-fp=filepath</b>                      | <a href="#">file_path</a>  | current directory  | Specifies location of database files.  |
| <b>-dp=datapath</b>                      | <a href="#">data_path</a>  | current directory  | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.   |
| <b>-nc=name_conv</b>                     | <a href="#">name_conv</a>  | name_conv          | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>                       | <a href="#">in_file</a>    | term_dat           | Specifies the input file name tag. The default is <b>term_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-ou=outout</b>                        | <a href="#">output</a>     | dec                | Specifies what type of output to be generated: <ul style="list-style-type: none"> <li>• <b>dec</b> -- prints the decoded information (default)</li> <li>• <b>code+dec</b> -- prints the raw TAF and decoded information</li> <li>• <b>eng</b> -- prints the decoded information in plain English</li> <li>• <b>code+eng</b> -- prints the raw TAF and plain English</li> </ul> |
| <b>-cf=cityfile</b>                      | <a href="#">city_file</a>  | None               | Specifies the name of the city database file used in finding station names.  |
| <b>-cu=[hour la]</b>                     | <a href="#">current</a>    | None               | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=hour</b>                          | <a href="#">hour</a>       | None               | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=num_hour</b>                      | <a href="#">num_hour</a>   | 0                  | This specifies the number of hours that will be searched for hourly data.  |
| <b>-id=identifier</b>                    | <a href="#">identifier</a> | User prompt        | Used to decode a specific TAF. The value " <b>all</b> " can be used to decode all reports. If not specified, the user is prompted for the ID.  |
| <b>-pa=param</b><br>[ <i>,param...</i> ] | <a href="#">parameter</a>  | None               | Extra parameters (none currently)  |
| <i>filename</i> [#seq]                   | <a href="#">filename</a>   | None<br>User Pompt | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-   |

|  |  |                                      |
|--|--|--------------------------------------|
|  |  | Batch: current= <b>la</b> WXP files. |
|--|--|--------------------------------------|

## DESCRIPTION

Terminal forecasts are coded in the format which is often hard to read and decode. As a result, it is handy to decode the TAFs into a readable form which can be used by the general public. By default, the program searches a single terminal forecast file (**term\_dat** in\_file type) and prompts the user for a station to decode. The data file is searched for that station and the appropriate TAF is decoded. A sample TAF is:

```
KORD 020525Z 020606 04012KT P6SM SCT120 OVC200
      BECMG 1416 04016G28KT P6SM OVC200
      BECMG 1820 04015G25KT P6SM BKN250
      BECMG 0002 05012KT P6SM SCT250=
```

The user may want to decode all the TAFs in a file. This can be done by specifying "**all**" for the station identifier either at the prompt or with the **identifier** resource. The default output is a reformatting into an easier to read form with some minimal decoding:

```
ID: KORD   CHICAGO/O'HARE, IL
REPORT TIME:  2  5:25
VALID TIME:  2 JUN AT 06Z TO 06Z NEXT DAY
INITIAL WX
  WINDS:  40 AT 12
  VISIBILITY:  >6.0 MILES
  CLOUDS:  120 SCT
  CLOUDS:  200 OVC
WX BECOMING FROM 14Z TO 16Z
  WINDS:  40 AT 16 GUST 28
  VISIBILITY:  >6.0 MILES
  CLOUDS:  200 OVC
WX BECOMING FROM 18Z TO 20Z
  WINDS:  40 AT 15 GUST 25
  VISIBILITY:  >6.0 MILES
  CLOUDS:  250 BKN
WX BECOMING FROM 00Z TO 02Z
  WINDS:  50 AT 12
  VISIBILITY:  >6.0 MILES
  CLOUDS:  250 SCT
COMMENTS:  =
```

If more decoding is desired, the output can be in a near English format with simple sentence structure:

```
Chicago/O'hare, IL (KORD) terminal forecast issued on 2 June at 05:25Z
valid from 2 June at 06Z to 06Z the next day.
```

```
The initial weather is:
winds from 40 deg at 12 kts, visibility is greater than 6.0 miles, clouds
12000 ft scattered, 20000 ft overcast.
```

```
The weather from 14Z to 16Z will change to:
winds from 40 deg at 16 kts with gusts to 28, visibility is greater than
6.0 miles, clouds 20000 ft overcast.
```

```
The weather from 18Z to 20Z will change to:
winds from 40 deg at 15 kts with gusts to 25, visibility is greater than
6.0 miles, clouds 25000 ft broken.
```

The weather from 00Z to 02Z will change to:  
winds from 50 deg at 12 kts, visibility is greater than 6.0 miles, clouds  
25000 ft scattered.

## FILES

- **term\_dat** - raw terminal forecast data files

## SEE ALSO

- [parse](#) - generic parsing program

# UA\_PARSE

## NAME

**ua\_parse** - Upper air data display program

## SYNOPSIS

**ua\_parse** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description  |
|------------------------------|----------------------------|-------------------|--|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.  |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wpxdef           | Sets the name of the resource file.  |
| <b>-na=name</b>              | <a href="#">name</a>       | ua_parse          | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode  |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• significant and mandatory sounding data - <b>out1a, out1b</b></li> <li>• standard sounding parameters and indices - <b>out1c, out1d</b></li> <li>• parcel information - <b>out2a, out2b</b></li> <li>• wind parameters - <b>out2c</b></li> <li>• wind level data - <b>out3a, out3b</b></li> <li>• wind/storm layer data - <b>out3c</b></li> </ul> |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | upa_cvt           | Specifies the input file name tag. The default is <b>upa_cvt</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | upa_snd           | Specifies what type of raw sounding file to create. By default, no output file is generated when the program. This specifies an output file name tag. The default is <b>upa_snd</b> . A full filename can be specified which can contain name convention tags.   |
| <b>-cu=[hour la]</b>         | <a href="#">current</a>    | None              | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=hour</b>              | <a href="#">hour</a>       | None              | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-id=identifier</b>        | <a href="#">identifier</a> | None              | Specifies the station to use in the program. Used to plot a specific observation. If not specified, all stations will be plotted.  |

|                                 |                            |  |  |
|---------------------------------|----------------------------|--|--|
| <b>-pp=param</b><br>[.param...] | <a href="#">plot_param</a> | None   | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.  |
| <b>filename[#seq]</b>           | <a href="#">filename</a>   | None<br>User Prompt<br>Batch: current= <b>la</b> | The name of the upper air converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. |

## DESCRIPTION

This program displays upper air sounding data. The input to the program is a upper air converted file produced by the [uacvt](#) programs.

The programs starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the [upa\\_cvt](#) tag but this can be changed with the [in\\_file](#) resource.

Next, the user enters a station identifier. A list of available stations from the converted file is displayed. The stations are listed by their 3/4 letter IDs and not the 5 digit WMO number for ease of selection. Once the station is selected, the program prints out sounding information, and standard sounding parameters and indices.

## Text Output

The first section is the station location information which is printed at message level **out1a**:

```
Date:0000Z 30 SEP 97
Station: ILX
WMO ident: 74560
Latitude: 40.15
Longitude: -89.33
Elevation: 178.00
```

The next section is the raw sounding data. This is a merged set of mandatory and significant level data. Additional computations are performed. Here is a list of the columns printed (all printed at mess level **out1a** [see red text] unless specified)

- LEV -- level number. Special levels are denoted with SFC (surface), TRP (tropopause), WND (max wind).
- PRES -- pressure of level in mb
- HGHT -- geopotential height in m (intermediate levels interpolated using hypsometric approximation)
- TEMP -- temperature in C
- DEWP -- dewpoint in C
- RH -- relative humidity in %
- DD -- dewpoint depression in C (mess level **out1b**)
- WETB -- wetbulb temperature in C (mess level **out1b**)
- DIR -- wind direction in deg (intermediate levels interpolated using significant wind level data)
- SPD -- wind speed in knt (intermediate levels interpolated using significant wind level data)
- THETA -- potential temperature in K (mess level **out1b**)
- THE-V -- virtual potential temperature in K (mess level **out1b**)
- THE-W -- wetbulb potential temperature in K (mess level **out1b**)
- THE-E -- equivalent potential temperature in K (mess level **out1b**)

| LEV | PRES | HGHT | TEMP | DEWP | RH | DD   | WETB | DIR | SPD | THETA | THE-V | THE-W | THE-E | W    |
|-----|------|------|------|------|----|------|------|-----|-----|-------|-------|-------|-------|------|
|     | mb   | m    | C    | C    | %  | C    | C    | deg | knt | K     | K     | K     | K     | g/kg |
| 0   | 1000 | -1   |      |      |    |      |      |     |     |       |       |       |       |      |
| SFC | 980  | 178  | 26.4 | 6.4  | 28 | 20.0 | 14.7 | 280 | 19  | 301.3 | 302.4 | 288.6 | 319.7 | 6.15 |
| 2   | 925  | 678  | 22.2 | 2.2  | 27 | 20.0 | 11.3 | 280 | 47  | 302.0 | 302.9 | 287.6 | 316.6 | 4.85 |
| 3   | 850  | 1403 | 15.2 | -0.8 | 33 | 16.0 | 7.1  | 280 | 46  | 302.1 | 302.8 | 287.1 | 315.0 | 4.24 |

```

4 700 3001 0.0 -6.0 64 6.0 -2.6 295 50 302.5 303.1 286.5 313.2 3.49
...

```

The next section displays standard sounding parameters and indices. Most of the information is printed at message level **out1d** except for thickness, lifted, Showalter, total totals and K indices which is at message level **out1c** (red text).

```

Sounding variables and indices
1000-500 mb thick:      5631.00 m
Freezing level:        700.00 mb = 3001.00 m = 9845.68 ft
Wetbulb zero:          740.60 mb = 2536.94 m = 8323.18 ft
Precipitable water:    0.65 inches
Sfc-500 mean rel hum:  36.73 %
Est. max temperature:  27.28 C = 81.10 F
Sfc-Lift cond lev (LCL): 729.89 mb = 2656.89 m = 8716.74 ft T: 3.27 C
700-500 lapse rate:   4.68 C/km
ThetaE index:          11.23 C Layer 925.0- 598.0 mb
Conv cond level (CCL): 689.45 mb = 3121.77 m = 10241.90 ft
  Mean mixing ratio:   5.11 g/kg
  Conv temperature:    27.64 C = 81.75 F
Cap Strength:          3.86 C
Lifted Index:          5.67 C Risk: None
Lifted Index @300 mb:  7.03 C
Lifted Index @700 mb:  0.59 C
Showalter Index:       6.96 C Risk: None
Total Totals Index:    39.00 C Risk: None
  Vertical Totals Index: 27.50 C
  Cross Totals Index:   11.50 C
K Index:               20.70 Risk: 20-40 % chance of thunderstorms
Sweat Index:           178.00 Risk: None
Energy Index:          1.12 Risk: None

```

The next section involves a parcel trajectory. The parcel is taken from some initial condition and raised vertically to the top of the sounding. The default parcel uses a 100 mb mean layer above the surface. The parcel type can be changed with the **par** plot parameter:

```
par=pres:temp:dew
```

Temperature and dewpoint are optional. If not specified, the sounding temperature and dewpoint are used.

| Type            | Pres  | Temp  | Dew |
|-----------------|-------|-------|-----|
| pressure level  | pres  | temp  | dew |
| surface level   | sfc   | temp  | dew |
| max temperature | maxt  | --    | --  |
| pressure layer  | layer | thick | --  |

The data are printed at message level **out2b** except for CAPE, convective inhibition and cap strength which is printed at level **out2a** (red text).

```

Parcel Indices
Parcel: using 100 mb layer
CAPE (B+):             10.89 J/kg
Max Up Vert Vel:       4.67 m/s
Conv Inhibition (B-):  30.45 J/kg
Cap Strength:          5.49 C
Lift Cond Lev (LCL):   696.21 mb = 3044.15 m = 9987.25 ft
Lev Free Conv (LFC):   681.21 mb = 3217.42 m = 10555.70 ft
Equ Level (EL):        631.21 mb = 3818.50 m = 12527.73 ft
B at Equ Level:        7.84 J/kg
Max Parcel Lev (MPL):  611.21 mb = 4071.09 m = 13356.44 ft

```

The next section displays significant wind level data. Storm relative winds are computed based on a mean storm motion. This is computed using the mean wind of the lower 6km of the atmosphere. The storm direction is deflected 30 deg clockwise and the speed is 75% of the 6km mean speed. The output is at message level **out3a** (red text) except for the storm relative winds which is at **out3b**.

```

Wind level data
Storm motion: 325 at 43 knt
-----

```

| LEV | P<br>mb | H<br>m | TRUE/GROUND REL |            |          |          | STORM REL |            |          |          |
|-----|---------|--------|-----------------|------------|----------|----------|-----------|------------|----------|----------|
|     |         |        | DIR             | SPD<br>knt | U<br>m/s | V<br>m/s | DIR       | SPD<br>knt | U<br>m/s | V<br>m/s |
| 0   | 980     | 178    | 280             | 19         | 18.7     | -3.3     | 170       | 32         | -5.6     | 31.9     |
| 1   | 966     | 305    | 280             | 30         | 29.5     | -5.2     | 190       | 30         | 5.3      | 30.0     |
| 2   | 932     | 610    | 275             | 46         | 45.8     | -4.0     | 215       | 38         | 21.6     | 31.2     |
| 3   | 925     | 678    | 280             | 47         | 46.3     | -8.2     | 219       | 35         | 22.0     | 27.0     |
| ... |         |        |                 |            |          |          |           |            |          |          |

The next section is wind parameters and indices. This is printed at message level **out2d** except for mean winds, storm direction, helicity and energy-helicity index which are at level **out2c** (red text)

```

Wind Parameters
Mean winds (0-6000m):           295.4 at 57.0 knts
Storm direction:                 325.4 at 42.8 knts
Shear (0-3000m)                  pos: 2.6 neg: 4.7 tot: 7.3 10^-3/s
Storm rel Dir Shear (0-3000m):   69.6 deg
Storm rel helicity (0-3000m) pos: 310.7 neg: -7.0 tot: 303.7 m^2/s^2
                                ave: 101.2 10^-3 m/s^2 rel: 0.84
Storm rel vorticity (0-3000m) horiz: 8.0 stream: 6.7 10^-3/s
Energy-Hel index:                 0.02
Bulk Rich Number:                 0.13
Bulk Shear:                       85.84 m/s

```

The next section shows storm layer parameters. All of this information is printed at message level **out3c**.  
Storm Parameters

```

-----

```

| Depth<br>AGL<br>m | Mean |     | Inflow |     | Shear   |      |      | Helicity |        |       |       | Vorticity |      |      |
|-------------------|------|-----|--------|-----|---------|------|------|----------|--------|-------|-------|-----------|------|------|
|                   | Dir  | Spd | Dir    | Spd | Pos     | Neg  | Tot  | Pos      | Neg    | Tot   | Ave   | Rel       | Hor  | Strm |
|                   | knt  | knt | knt    | knt | 10^-3/s |      |      | m^2/s^2  |        |       |       | 10^-3/s   |      |      |
| 500               | 278  | 36  | 200    | 32  | 0.0     | 28.8 | 28.8 | 225.6    | 0.0    | 225.6 | 451.2 | 1.00      | 29.7 | 29.6 |
| 1000              | 279  | 41  | 210    | 33  | 0.0     | 14.0 | 14.0 | 225.6    | -7.0   | 218.6 | 218.6 | 0.94      | 15.3 | 14.4 |
| 1500              | 280  | 43  | 213    | 33  | 0.9     | 9.3  | 10.2 | 229.6    | -7.0   | 222.6 | 148.4 | 0.88      | 11.1 | 9.8  |
| 2000              | 281  | 44  | 215    | 33  | 1.1     | 7.0  | 8.1  | 249.4    | -7.0   | 242.5 | 121.2 | 0.89      | 9.0  | 7.9  |
| 2500              | 282  | 45  | 217    | 32  | 1.7     | 5.6  | 7.3  | 280.2    | -7.0   | 273.2 | 109.3 | 0.89      | 8.0  | 7.1  |
| 3000              | 284  | 45  | 219    | 31  | 2.6     | 4.7  | 7.3  | 310.7    | -7.0   | 303.7 | 101.2 | 0.84      | 8.0  | 6.7  |
| 4000              | 289  | 47  | 225    | 28  | 2.8     | 4.6  | 7.3  | 351.2    | -7.0   | 344.3 | 86.1  | 0.75      | 7.9  | 6.0  |
| 5000              | 294  | 51  | 237    | 26  | 3.3     | 4.6  | 7.9  | 458.4    | -7.0   | 451.4 | 90.3  | 0.73      | 8.5  | 6.2  |
| 6000              | 295  | 57  | 248    | 29  | 2.7     | 5.6  | 8.4  | 458.4    | -100.4 | 358.0 | 59.7  | 0.50      | 9.0  | 4.5  |

## EXAMPLES

```
ua_parse -cu=la -id=ILX -de=d
```

This displays sounding data for ILX.

## FILES

- **upa.cty** - the surface station location database file

## SEE ALSO

- [uacvt](#) - the upper air decoding program
- [uacalplt](#) - the upper air sounding plotting program

# UACALPLT

## NAME

**uacalplt** - Upper air calculation and plotting program

## SYNOPSIS

**uacalplt** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description  |
|------------------------------|----------------------------|-------------------|--|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.  |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxpdef           | Sets the name of the resource file.  |
| <b>-na=name</b>              | <a href="#">name</a>       | uacalplt          | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode  |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• significant and mandatory sounding data - <b>out1a, out1b</b></li> <li>• standard sounding parameters and indices - <b>out1c, out1d</b></li> <li>• parcel information - <b>out2a, out2b</b></li> <li>• wind parameters - <b>out2c</b></li> <li>• wind level data - <b>out3a, out3b</b></li> <li>• wind/storm layer data - <b>out3c</b></li> </ul> |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.  |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | upa_cvt           | Specifies the input file name tag. The default is <b>upa_cvt</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | upa_snd           | Specifies what type of raw sounding file to create. By default, no output file is generated when the program. This specifies an output file name tag. The default is <b>upa_snd</b> . A full filename can be specified which can contain name convention tags.   |
| <b>-cu=[hour la]</b>         | <a href="#">current</a>    | None              | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho=hour</b>              | <a href="#">hour</a>       | None              | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-id=identifier</b>        | <a href="#">identifier</a> | None              | Specifies the station to use in the program. Used to plot a  |

|                                 |                               |  |  |
|---------------------------------|-------------------------------|--|--|
|                                 |                               |  | specific observation. If not specified, all stations will be plotted.  |
| <b>-pd=domain</b>               | <a href="#">plot_domain</a>   | None.<br>Full plot   | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-pl=plot_type</b>            | <a href="#">plot_type</a>     | User prompt<br>Batch: <b>skewt</b>                         | Specifies the type of output thermodynamic plot. By default, this is a Skew T diagram. The available plot types are: <ul style="list-style-type: none"> <li>• <b>skewt</b> -- Skew T log P diagram</li> <li>• <b>ema</b> -- Emagram</li> <li>• <b>stuve</b> -- Stuve diagram</li> <li>• <b>hodo</b> -- Hodogram</li> <li>• <b>none</b> -- for no output plot</li> </ul>  |
| <b>-sc=scale_factor</b>         | <a href="#">plot_scale</a>    | <b>1.0</b>   | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm=format</b>               | <a href="#">plot_format</a>   | None   | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp=param</b><br>[,param...] | <a href="#">plot_param</a>    | None   | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.  |
| <b>-dr=drawable[,...]</b>       | <a href="#">draw</a>          | <b>all</b>   | Specifies which part of the plot is to be drawn.   |
| <b>-la=label[,...]</b>          | <a href="#">label</a>         | Program defaults   | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-fl=font_list</b>            | <a href="#">font_list</a>     | <b>modern.fnt</b>  | Specifies a list of fonts to use in the plots.   |
| <b>-ct=color_table</b>          | <a href="#">color_table</a>   | <b>wxp.clr</b>   | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |
| <b>-cot=color[:attr]</b>        | <a href="#">color_text</a>    | <b>white</b><br><b>black</b> for inverse plots.            | Specifies the text color for labels above and below the plot.  |
| <b>-coth=color[:attr]</b>       | <a href="#">color_therm</a>   | <b>blue, blue,</b><br><b>green, cyan,</b><br><b>yellow</b> | This resource is used to specify the color and attributes for the thermodynamic plot. There are 5 possible values: <ul style="list-style-type: none"> <li>• pressure lines (default <b>blue</b>)</li> <li>• temperature lines (default <b>blue</b>)</li> <li>• potential temperature lines -- dry adiabats (default <b>green</b>)</li> <li>• equivalent potential temperature lines -- moist adiabats (default <b>cyan</b>)</li> <li>• mixing ratio lines (default <b>yellow</b> or <b>brown</b> on inverse color plots)</li> </ul>  |
| <b>-cott=color[:attr]</b>       | <a href="#">color_thrmtxt</a> | <b>lblue, lblue, off,</b><br><b>off, yellow</b>            | This resource is used to specify the color and attributes for the labels on the thermodynamic plot. There are 5 possible values: <ul style="list-style-type: none"> <li>• pressure lines (default <b>blue</b>)</li> <li>• temperature lines (default <b>blue</b>)</li> <li>• potential temperature lines -- dry adiabats (default <b>off</b> - does not plot normally)</li> <li>• equivalent potential temperature lines -- moist adiabats (default <b>off</b> - does not plot normally)</li> <li>• mixing ratio lines (default <b>yellow</b> or <b>brown</b> on inverse color plots)</li> </ul> |
| <b>-cosd=color[:attr]</b>       | <a href="#">color_sound</a>   | <b>white, white,</b><br><b>yellow</b>                      | This resource is used to specify the color and attributes for the sounding plot. There are 3 possible values: <ul style="list-style-type: none"> <li>• temperature sounding (default <b>white</b> or <b>black</b> for inverse plots)</li> <li>• dewpoint sounding (default <b>white</b> or <b>black</b> for inverse plots)</li> </ul>  |

|                                    |                              |   |   |
|------------------------------------|------------------------------|---|---|
|                                    |                              |   | <ul style="list-style-type: none"> <li>parcel sounding (default <b>yellow</b> or <b>brown</b> on inverse color plots)</li> </ul>  |
| <b>-c</b> <i>owd=</i> color[:attr] | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse plots. | Specifies the color for wind barbs on thermodynamic plots.  |
| <b>-c</b> <i>oho=</i> color[:attr] | <a href="#">color_hodo</a>   | <b>blue, blue,</b><br><b>white, cyan, red</b>   | This resource is used to specify the color and attributes for the hodograph plot. There are 5 possible values: <ul style="list-style-type: none"> <li>direction lines (default <b>blue</b>)</li> <li>speed circles (default <b>blue</b>)</li> <li>sounding line up to 700 mb (default <b>white</b> or <b>black</b> for inverse plots)</li> <li>sounding line above 700 mb (default <b>cyan</b>)</li> <li>marker "+" denoting each level (default <b>red</b>)</li> </ul> |
| <b>-c</b> <i>oht=</i> color[:attr] | <a href="#">color_hodtxt</a> | <b>lblue, lblue, lred</b>                       | This resource is used to specify the color and attributes for the labels on the hodograph plot. There are 3 possible values: <ul style="list-style-type: none"> <li>direction labels (default <b>lblue</b>)</li> <li>speed labels (default <b>lblue</b>)</li> <li>sounding labels denoting pressure level (default <b>lred</b>)</li> </ul>  |
| <b>-d</b> <i>e=</i> device[,...]   | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>            | Specifies the output device.  |
| <b>-t</b> <i>i=</i> title          | <a href="#">title</a>        | Program name                                    | Specifies the window title.   |
| <b>-i</b> <i>c</i>                 | <a href="#">icon</a>         | <b>Off</b>                                      | Specifies whether to open window in iconified mode.   |
| <b>-b</b> <i>g=</i> color          | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse plots  | Specifies the background color of the window.   |
| <b>-g</b> <i>e=</i> geometry       | <a href="#">geometry</a>     | <b>640x480</b>                                  | Specifies the window and paper geometry.  |
| <i>f</i> ilename[#seq]             | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the upper air converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files.  |

## DESCRIPTION

This program plots upper air sounding data on a thermodynamic diagram or hodograph. The input to the program is an upper air converted file produced by the [uacvt](#) programs.

The program starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **upa\_cvt** tag but this can be changed with the [in\\_file](#) resource.

Next, the user enters a station identifier. A list of available stations from the converted file is displayed. The stations are listed by their 3/4 letter IDs and not the 5 digit WMO number for ease of selection. Once the station is selected, the program prints out sounding information, and standard sounding parameters and indices.

### Text Output

The first section is the station location information which is printed at message level **out1a**:

```
Date:0000Z 30 SEP 97
Station: ILX
WMO ident: 74560
Latitude: 40.15
Longitude: -89.33
Elevation: 178.00
```

The next section is the raw sounding data. This is a merged set of mandatory and significant level data. Additional computations are performed. Here is a list of the columns printed (all printed at mess level **out1a** [see red text] unless specified)

- LEV -- level number. Special levels are denoted with SFC (surface), TRP (tropopause), WND (max wind).
- PRES -- pressure of level in mb
- HGHT -- geopotential height in m (intermediate levels interpolated using hypsometric approximation)
- TEMP -- temperature in C
- DEWP -- dewpoint in C
- RH -- relative humidity in %
- DD -- dewpoint depression in C (mess level **out1b**)
- WETB -- wetbulb temperature in C (mess level **out1b**)
- DIR -- wind direction in deg (intermediate levels interpolated using significant wind level data)
- SPD -- wind speed in knt (intermediate levels interpolated using significant wind level data)
- THETA -- potential temperature in K (mess level **out1b**)
- THE-V -- virtual potential temperature in K (mess level **out1b**)
- THE-W -- wetbulb potential temperature in K (mess level **out1b**)
- THE-E -- equivalent potential temperature in K (mess level **out1b**)

```

-----
LEV PRES  HGHT  TEMP  DEWP  RH  DD  WETB  DIR  SPD  THETA  THE-V  THE-W  THE-E  W
      mb    m    C    C    %  C    C    deg  knt  K      K      K      K      g/kg
-----
  0 1000   -1
SFC 980   178  26.4   6.4  28 20.0  14.7  280  19 301.3  302.4  288.6  319.7  6.15
  2  925   678  22.2   2.2  27 20.0  11.3  280  47 302.0  302.9  287.6  316.6  4.85
  3  850  1403  15.2  -0.8  33 16.0   7.1  280  46 302.1  302.8  287.1  315.0  4.24
  4  700  3001   0.0  -6.0  64  6.0  -2.6  295  50 302.5  303.1  286.5  313.2  3.49
...

```

The next section displays standard sounding parameters and indices. Most of the information is printed at message level **out1d** except for thickness, lifted, Showalter, total totals and K indices which is at message level **out1c** (red text).

```

Sounding variables and indices
1000-500 mb thick:      5631.00 m
Freezing level:        700.00 mb = 3001.00 m = 9845.68 ft
Wetbulb zero:          740.60 mb = 2536.94 m = 8323.18 ft
Precipitable water:    0.65 inches
Sfc-500 mean rel hum:  36.73 %
Est. max temperature:  27.28 C = 81.10 F
Sfc-Lift cond lev (LCL): 729.89 mb = 2656.89 m = 8716.74 ft T: 3.27 C
700-500 lapse rate:   4.68 C/km
ThetaE index:          11.23 C Layer 925.0- 598.0 mb
Conv cond level (CCL): 689.45 mb = 3121.77 m = 10241.90 ft
  Mean mixing ratio:    5.11 g/kg
  Conv temperature:     27.64 C = 81.75 F
Cap Strength:          3.86 C
Lifted Index:          5.67 C Risk: None
Lifted Index @300 mb:  7.03 C
Lifted Index @700 mb:  0.59 C
Showalter Index:       6.96 C Risk: None
Total Totals Index:    39.00 C Risk: None
  Vertical Totals Index: 27.50 C
  Cross Totals Index:   11.50 C
K Index:               20.70 Risk: 20-40 % chance of thunderstorms
Sweat Index:           178.00 Risk: None
Energy Index:          1.12 Risk: None

```

The next section involves a parcel trajectory. The parcel is taken from some initial condition and raised vertically to the top of the sounding. The default parcel uses a 100 mb mean layer above the surface. The parcel type can be changed with the **par** plot parameter:

**par=**pres:temp:dew

Temperature and dewpoint are optional. If not specified, the sounding temperature and dewpoint are used.

| Type            | Pres         | Temp         | Dew        |
|-----------------|--------------|--------------|------------|
| pressure level  | <b>pres</b>  | <i>temp</i>  | <i>dew</i> |
| surface level   | <b>sfc</b>   | <i>temp</i>  | <i>dew</i> |
| max temperature | <b>maxt</b>  | --           | --         |
| pressure layer  | <b>layer</b> | <i>thick</i> | --         |

The data are printed at message level **out2b** except for CAPE, convective inhibition and cap strength which is printed at level **out2a** (red text).

```
Parcel Indices
Parcel: using 100 mb layer
CAPE (B+):          10.89 J/kg
Max Up Vert Vel:    4.67 m/s
Conv Inhibition (B-): 30.45 J/kg
Cap Strength:       5.49 C
Lift Cond Lev (LCL): 696.21 mb = 3044.15 m = 9987.25 ft
Lev Free Conv (LFC): 681.21 mb = 3217.42 m = 10555.70 ft
Equ Level (EL):     631.21 mb = 3818.50 m = 12527.73 ft
B at Equ Level:     7.84 J/kg
Max Parcel Lev (MPL): 611.21 mb = 4071.09 m = 13356.44 ft
```

The next section displays significant wind level data. Storm relative winds are computed based on a mean storm motion. This is computed using the mean wind of the lower 6km of the atmosphere. The storm direction is deflected 30 deg clockwise and the speed is 75% of the 6km mean speed. The output is at message level **out3a** (red text) except for the storm relative winds which is at **out3b**.

```
Wind level data
Storm motion: 325 at 43 knt
-----
      LEV   P   H   TRUE/GROUND REL   |   STORM REL
      DIR SPD   U   V   DIR SPD   U   V
      mb    m   knt  m/s   knt    m/s
-----|-----
      0  980  178  280 19 18.7 -3.3 | 170 32 -5.6 31.9
      1  966  305  280 30 29.5 -5.2 | 190 30  5.3 30.0
      2  932  610  275 46 45.8 -4.0 | 215 38 21.6 31.2
      3  925  678  280 47 46.3 -8.2 | 219 35 22.0 27.0
      ...
```

The next section is wind parameters and indices. This is printed at message level **out2d** except for mean winds, storm direction, helicity and energy-helicity index which are at level **out2c** (red text)

```
Wind Parameters
Mean winds (0-6000m):          295.4 at 57.0 knts
Storm direction:              325.4 at 42.8 knts
Shear (0-3000m) pos:          2.6 neg: 4.7 tot: 7.3 10^-3/s
Storm rel Dir Shear (0-3000m): 69.6 deg
Storm rel helicity (0-3000m) pos: 310.7 neg: -7.0 tot: 303.7 m^2/s^2
ave: 101.2 10^-3 m/s^2 rel: 0.84
Storm rel vorticity (0-3000m) horiz: 8.0 stream: 6.7 10^-3/s
```

```

Energy-Hel index:      0.02
Bulk Rich Number:     0.13
Bulk Shear:           85.84 m/s

```

The next section shows storm layer parameters. All of this information is printed at message level **out3c**.

#### Storm Parameters

| Depth<br>AGL<br>m | Mean |     | Inflow |     | Shear               |      |      | Helicity                       |        |       |                  | Vorticity |                     |      |
|-------------------|------|-----|--------|-----|---------------------|------|------|--------------------------------|--------|-------|------------------|-----------|---------------------|------|
|                   | Dir  | Spd | Dir    | Spd | Pos                 | Neg  | Tot  | Pos                            | Neg    | Tot   | Ave              | Rel       | Hor                 | Strm |
|                   | knt  | knt |        |     | 10 <sup>-3</sup> /s |      |      | m <sup>2</sup> /s <sup>2</sup> |        |       | m/s <sup>2</sup> |           | 10 <sup>-3</sup> /s |      |
| 500               | 278  | 36  | 200    | 32  | 0.0                 | 28.8 | 28.8 | 225.6                          | 0.0    | 225.6 | 451.2            | 1.00      | 29.7                | 29.6 |
| 1000              | 279  | 41  | 210    | 33  | 0.0                 | 14.0 | 14.0 | 225.6                          | -7.0   | 218.6 | 218.6            | 0.94      | 15.3                | 14.4 |
| 1500              | 280  | 43  | 213    | 33  | 0.9                 | 9.3  | 10.2 | 229.6                          | -7.0   | 222.6 | 148.4            | 0.88      | 11.1                | 9.8  |
| 2000              | 281  | 44  | 215    | 33  | 1.1                 | 7.0  | 8.1  | 249.4                          | -7.0   | 242.5 | 121.2            | 0.89      | 9.0                 | 7.9  |
| 2500              | 282  | 45  | 217    | 32  | 1.7                 | 5.6  | 7.3  | 280.2                          | -7.0   | 273.2 | 109.3            | 0.89      | 8.0                 | 7.1  |
| 3000              | 284  | 45  | 219    | 31  | 2.6                 | 4.7  | 7.3  | 310.7                          | -7.0   | 303.7 | 101.2            | 0.84      | 8.0                 | 6.7  |
| 4000              | 289  | 47  | 225    | 28  | 2.8                 | 4.6  | 7.3  | 351.2                          | -7.0   | 344.3 | 86.1             | 0.75      | 7.9                 | 6.0  |
| 5000              | 294  | 51  | 237    | 26  | 3.3                 | 4.6  | 7.9  | 458.4                          | -7.0   | 451.4 | 90.3             | 0.73      | 8.5                 | 6.2  |
| 6000              | 295  | 57  | 248    | 29  | 2.7                 | 5.6  | 8.4  | 458.4                          | -100.4 | 358.0 | 59.7             | 0.50      | 9.0                 | 4.5  |

## Sounding Plots

After the text is displayed, the user is prompted for the output plot type. The selections are:

- **skewt** -- Skew T log P diagram
- **ema** -- Emagram
- **stuve** -- Stuve diagram
- **hodo** -- Hodogram

A thermodynamic diagram is plot showing the temperature and dewpoint plotted versus pressure (or height). A hodograph is a plot of wind speed versus direction on a polar grid.

**Skew-T log-P:** The skew-T log-P diagram is an energy conserving diagram in which the temperature and potential temperature lines are nearly perpendicular. The pressure lines are plotted horizontally in blue and are also on an inverse log scale. The concept of Skew T means that the temperature is not plotted vertically but angles off to the right at a 45 degree angle. The temperature lines of the Skew T are in blue. The dry adiabats slope off to the upper left and are plotted in green. The light blue dashed curved lines are saturation adiabats. The yellow dashed lines are lines of constant mixing ratio.

The sounding is displayed in white, with the white line on the left of the plot representing the dewpoint line and the white data line on the right representing the temperature line. When printed, these are bold lines for easier viewing. The wind barbs on the right-hand side of the diagram represent the winds at various elevations, with the top of the diagram representing north. The winds at mandatory levels are plotted at their pressure levels, whereas the winds at height levels are plotted at their standard atmosphere pressure level.

**Emagram:** An emagram is very close to a skewT except the temperature lines are vertical, not skewed to the right. It is also an energy conserving diagram.

**Stuve:** In the Stuve diagram format, the diagram is much the same as the Skew T plot except that the temperature lines are vertical. Pressure lines are spaced so that potential temperature lines are straight. This is not an energy conserving diagram.

**Wind plot:** The winds at selected levels are plotted on the right side of the diagram. Standard directions are used with up being north and right being to the east. These are plotted in white but can be changed with the **color\_wind** resource.

**Data plot:** The geopotential heights are plotted at a 50 mb interval on the left side of the plot. The color of the data can be changed with the **color\_wind** resource. On the right side of the plot, specific parameters and indices are plotted. These values are:

Station information:

|            |                                |
|------------|--------------------------------|
| <b>WMO</b> | WMO number                     |
| <b>LAT</b> | Latitude of station            |
| <b>LON</b> | Longitude of station           |
| <b>EL</b>  | Elevation of station in meters |

Key information:

|             |   |
|-------------|---|
| <b>TP</b>   | Tropopause level (mb)                               |
| <b>MW</b>   | Max wind level (mb)                                 |
| <b>FRZ</b>  | Lowest freezing level (mb) or BG for below ground   |
| <b>WB0</b>  | Wet bulb zero (mb) or BG                            |
| <b>PW</b>   | Precipitable water (in)                             |
| <b>RH</b>   | Mean RH surface to 500 mb (%)                       |
| <b>MAXT</b> | Estimated max temperature (C) using a 150mb layer   |
| <b>TH</b>   | 1000-500mb thickness (m)                            |
| <b>L57</b>  | 700-500mb lapse rate (C/km)                         |
| <b>LCL</b>  | Lift condensation level (mb) from surface data      |
| <b>LI</b>   | Lifted index (C) using 100 mean layer above surface |
| <b>SI</b>   | Showalter index (C)                                 |
| <b>TT</b>   | Total totals index                                  |
| <b>KI</b>   | K index   |
| <b>SW</b>   | Sweat index   |
| <b>EI</b>   | Energy index  |

#### **-PARCEL-**

This is a parcel trajectory (the yellow line on the sounding) based on 100 mb mean layer.

|             |                                       |
|-------------|---------------------------------------|
| <b>CAPE</b> | Convective available potential energy |
| <b>CINH</b> | Convective inhibition (open ended)    |
| <b>LCL</b>  | Lift condensation level (mb)          |
| <b>CAP</b>  | Cap strength (C)                      |
| <b>LFC</b>  | Level free convection (mb)            |
| <b>EL</b>   | Equilibrium level (mb)                |
| <b>MPL</b>  | Maximum parcel level (mb)             |

#### **-WIND-**

Wind parameters

|             |   |
|-------------|---|
| <b>STM</b>  | Estimated storm motion (knts) from 0-6000m AG mean layer, spd 75% of mean, dir 30 deg veer from mean. |
| <b>HEL</b>  | Storm relative helicity 0-3000m AG (total value)  |
| <b>SHR+</b> | Positive shear magnitude 0-3000m AG (sum of veering shear values)                                     |
| <b>SRDS</b> | Storm relative directional shear 0-3000m AG (directional difference of storm relative winds)          |
| <b>EHI</b>  | Energy helicity index (prop to positive helicity * CAPE)  |
| <b>BRN</b>  | Bulk Richardson number 500-6000m AG (prop to CAPE/bulk shear)   |
| <b>BSHR</b> | Bulk shear value (magnitude of shear over layer)  |

The amount of data plotted can be regulated by using the **data** plot parameters:

- **nodata** -- no data plotted
- **data** -- all data plotted

- **data1** -- minimal data plotted
- **data2** -- important data plotted

**Parcel Trajectory:** A parcel trajectory is plotted with each sounding as a thick yellow line. If no parcel trajectory is needed, specifying the **noparcel** plot parameter will eliminate it from the plot. The type of parcel can be changed by specifying the **par** plot parameter.

**Wind hodograph:** The wind hodograph is a polar plot of wind speed and direction. The radius directly reflects wind speed. The hodograph is plotted for all mandatory and wind at height levels with the plot annotated with plus signs at each level and labeled with the mandatory level pressures above and to the right of the plus sign. The lowest level data (below 700 mb) is plotted in white.

The plot type can be changed. If the thermodynamic diagram is visible, the hodograph can be plotted by hitting <Meta>n. By hitting <Meta>n again, the thermodynamic plot will reappear.

### Sounding Modification

Once the data are plotted, the sounding can be modified. This can be done by moving the cursor to the line which is to be modified, holding down the <Shift-right button> and moving the cursor to where the new value will be. If the cursor is right of the temperature line, the temperature sounding is modified. If the cursor is left of the dewpoint line, the dewpoint is modified. This can only be done on the existing levels. This can also be done on the hodograph.

In many cases, the sounding has too much detail to effectively select a level or data point. It is recommended that the user zoom the plot by using the <Shift-left button> and dragging a box around the area to be zoomed. Remember to start at a corner.

Once the sounding is modified, the plotted data/parameters/indices will update to reflect the changed sounding. The parcel will not automatically update. This can be done with the <Meta>r refresh command. The modified sounding can be saved to a raw file using the <Meta>s save command. This can be read in by **uacalplt** to be replotted.

### Output Files

By default, this program does not produce output files. A raw file can be generated using the <Meta>s command in the plot window. The **out\_file** resource specifies a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

### EXAMPLES

```
uacalplt -cu=la -id=ILX -pl=skewt -de=d
```

This plots a skewT diagram for ILX.

### FILES

- **upa.cty** - the surface station location database file

### SEE ALSO

- [uacvt](#) - the upper air decoding program

# UACVT

## NAME

**uacvt** - Upper air data conversion program

## SYNOPSIS

**uacvt** [*parameters...*] *filename*

## PARAMETERS

| Command Line          | Resource                   | Default                          | Description   |
|-----------------------|----------------------------|----------------------------------|---|
| <b>-h</b>             | <a href="#">help</a>       | No                               | Lists basic help information.   |
| <b>-df=filename</b>   | <a href="#">default</a>    | .wxpdef                          | Sets the name of the resource file.   |
| <b>-na=name</b>       | <a href="#">name</a>       | <b>uacvt</b>                     | Specifies the name used in resource file parsing.   |
| <b>-ba</b>            | <a href="#">batch</a>      | No                               | Run program in batch mode   |
| <b>-me=level</b>      | <a href="#">message</a>    | <b>out2</b>                      | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• station information - <b>out3a</b></li> <li>• not in database - <b>out3c</b></li> <li>• bad date information - <b>out3c</b></li> <li>• decoded report - <b>out4a</b></li> <li>• raw data and format - <b>out4c</b></li> </ul>  |
| <b>-fp=filepath</b>   | <a href="#">file_path</a>  | current directory                | Specifies location of database files.   |
| <b>-dp=datapath</b>   | <a href="#">data_path</a>  | current directory                | Specifies the location (path) of the input raw data files. This is the location where the ingest program has saved the data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>    | <a href="#">con_path</a>   | current directory                | Specifies the location (path) of the output converted data files. This may be modified in the name convention file.   |
| <b>-nc=name_conv</b>  | <a href="#">name_conv</a>  | <b>name_conv</b>                 | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>    | <a href="#">in_file</a>    | <b>upa_dat</b>                   | Specifies the input file name tag. The default is <b>upa_dat</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-ou=output</b>     | <a href="#">output</a>     | <b>wxp</b>                       | Specifies what type of output file is to be created: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII converted file. Name convention set to <b>upa_cvt_wxp</b>.</li> <li>• <b>cdf</b> - NetCDF binary converted file. Name convention set to <b>upa_cvt_cdf</b>.</li> </ul>   |
| <b>-of=out_file</b>   | <a href="#">out_file</a>   | <b>upa_cvt</b>                   | The <b>out_file</b> resource specifies the output file name tag. The default is <b>upa_cvt</b> but it can be modified to any value in the filename convention file. A full name convention can be specified as well.<br>Also, can specify what type of output file is to be created: <ul style="list-style-type: none"> <li>• <b>wxp</b> - WXP ASCII converted file</li> <li>• <b>cdf</b> - NetCDF binary converted file</li> </ul> |
| <b>-pf=file_param</b> | <a href="#">file_param</a> | User prompt<br>Batch: <b>use</b> | This specifies how to treat files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> </ul>   |

|   |                             |   |  |
|---|-----------------------------|---|--|
|   |                             |   | <ul style="list-style-type: none"> <li>• <b>N</b> - same as use.</li> </ul>  |
| <b>-cf=cityfile</b>                                 | <a href="#">city file</a>   | <b>upa.cty</b>                                  | Specifies the name of the city database file used for station lookup.<br><b>NOTE:</b> Stations must exist in database file in order to be decoded.   |
| <b>-cu=[hour]la</b>                                 | <a href="#">current</a>     | None  | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=hour</b>                                     | <a href="#">hour</a>        | None  | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=num_hour</b>                                 | <a href="#">num_hour</a>    | <b>0</b>  | This specifies the number of hours that will be searched for hourly data.  |
| <b>-dh=decode_hour</b>                              | <a href="#">decode_hour</a> | None  | Specifies a specific decode time when using more than one file.  |
| <b>-id=identifier</b>                               | <a href="#">identifier</a>  | None  | Used to decode a specific observation. If not specified, all stations will be decoded.   |
| <b>-pa=param</b><br>[,param...]                     | <a href="#">parameter</a>   | None  | Extra parameters: <ul style="list-style-type: none"> <li>• <b>range:pre:post</b> - specifies time range for valid data. (default: -120:120)</li> </ul>   |
| <i>filename[#seq]</i><br><i>[filename[#seq]...]</i> | <a href="#">filename</a>    | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the raw data file to be converted. An optional sequence number can be added to designate the time for non-Batch: <b>current=la</b> WXP files. More than one file can be specified.   |

## DESCRIPTION

The intent of the upper air conversion process is to collate the 6 types of upper air reports into one report for ease of use. The six products include:

- **TTAA** - mandatory level data up to 100 mb (sfc, 1000, 925, 850, 700, 500, 400, 300, 250, 200, 150, 100)
- **TTCC** - mandatory level data above 100 mb (70, 50, 30, 20, 10, 7, 5...)
- **TTBB** - significant level data up to 100 mb
- **TTDD** - significant level data above 100 mb
- **PPAA** - mandatory wind level data up to 100 mb
- **PPBB** - significant wind level data up to 50000 feet
- **PPDD** - significant wind level data above 50000 feet.

The program will first prompt the user for the input raw data file. This file should contain the upper air reports for a given time. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **upa\_dat** tag but this can be changed with the [in\\_file](#) resource. If the data stretches over an hour, the [num\\_hour](#) resource can be specified to search over multiple files.

Once the input file is determined, the program determines the decode time from the filename specified or that of the **current** resource. If decode time is different from that of the specified file, the [decode\\_hour](#) resource must be specified. This resource can also be used to set decode time range limits which is useful if decoding data from off times such as 90 after the hour.

The program will then read in the **upa.cty** database file in order to generate a database of upper air stations. The program When one of the above products is found, it is decoded and the resulting information placed into the database entry for that station. When all the data files have been searched and decoded, the output is saved to a converted file. The station data is saved with its 3 or 4 letter ICAO identifier for consistency with the other data types. The name convention is determined from the **upa\_cvt** file convention tag. If the converted file exists, the user will be prompted whether to overwrite, append or use the existing file. The append option leaves the current data in the converted file and adds the new station data onto the end of the file. If the use option is specified, the converted file is not modified and the surface data conversion program exits.

The conversion process is silent. By default, none of the output is displayed to the screen as an attempt to speed up the conversion process. A list of decoded stations can be displayed if the **out3** message level is set:

```
Station not in database: 76805
Saving stations...
STA: BRW WMO: 70026 LAT: 71.30 LON: -156.78
STA: OTZ WMO: 70133 LAT: 66.87 LON: -162.63
STA: OME WMO: 70200 LAT: 64.50 LON: -165.43
STA: BET WMO: 70219 LAT: 60.78 LON: -161.80
STA: FAI WMO: 70261 LAT: 64.82 LON: -147.87
STA: ANC WMO: 70273 LAT: 61.17 LON: -150.02
```

Any stations that were in the data file but not decoded are also listed. It is recommended that these stations be added to the database.

There are two types of output data file types available. The ASCII wxp Flat file data type saves all available information in a columnar format which may be edited with a standard text editor. The netCDF data format is a binary format which may be produced by specifying **cdf** for the output file type.

**NOTE:** **uacvt** must be run before any upper air plotting and gridding programs may be run.

## EXAMPLES

```
uacvt -cu=la -pf=over -me=print
```

This converts the latest upper air data file and overwrites any existing file.

```
uacvt -cu=la -dh=18 -ho=12 -of=upa3_cvt
```

This will decode 18Z data from WXP ingested data files. Since the data from the WXP ingestor is saved in 12 hourly files, the **hour** resource is used to specify which hourly file to use. The **decode\_hour** resource tells the program to decode the 18Z data in that file. The output file type is set to **upa3\_cvt** in order to set the output file type to a 3 hourly converted file. Otherwise the hour will be rounded down to 12Z.

The equivalent from the LDM is:

```
uacvt -ho=18 -cu=la -of=upa3_cvt
```

which is much simpler since data is stored by hour.

## FILES

- **upa.cty** - the upper air station location database file

## SEE ALSO

- [upairwx](#) - the surface data plotting program
- [upcalc](#) - the surface data contouring program
- [uacalplt](#) - the surface meteogram plotting program

# UNIT

## NAME

**unit** - Unit conversion program

## SYNOPSIS

**unit** [*parameters*] [*value old\_unit new\_unit*]

## PARAMETERS

| Command Line         | Resource                  | Default        | Description  |
|----------------------|---------------------------|----------------|--|
| <b>-h</b>            | <a href="#">help</a>      | No             | Lists basic help information.  |
| <b>-df=filename</b>  | <a href="#">default</a>   | <b>.wxddef</b> | Sets the name of the resource file.  |
| <b>-na=name</b>      | <a href="#">name</a>      | <b>heat</b>    | Specifies the name used in resource file parsing.  |
| <b>-ba</b>           | <a href="#">batch</a>     | No             | Run program in batch mode  |
| <b>-me=level</b>     | <a href="#">message</a>   | <b>out2</b>    | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• output values - <b>print</b></li> <li>• value labels - <b>mess</b></li> <li>• table -<b>out1</b></li> </ul> |
| <b>-pa=parameter</b> | <a href="#">parameter</a> | none           | Specifies additional parameters  |
| <i>value</i>         | <a href="#">value</a>     | none           | Specifies the input value  |
| <i>old_unit</i>      |                           | User prompt    | Specifies the units of the input value   |
| <i>new_unit</i>      |                           |                | Specifies the new units  |

## DESCRIPTION

The unit program is an interface into the WXP units conversion library. The user specifies an input value and its units plus a requested output units. The program will then search the **units.lup** file for an appropriate conversion. If a conversion is found, the result is displayed. Otherwise the initial value is displayed.

```
unit 30 F C
-1.111270
```

## FILES

- **units.lup** - The units conversion lookup file.

## SEE ALSO

# UPAIRWX

## NAME

**upairwx** - Upper air data plotting program

## SYNOPSIS

**upairwx** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description   |
|------------------------------|----------------------------|-------------------|---|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.   |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxpdef           | Sets the name of the resource file.   |
| <b>-na=name</b>              | <a href="#">name</a>       | upairwx           | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode   |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>gridding information - <b>out3</b></li> </ul>  |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.   |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.  |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.   |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.   |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.   |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | upa_cvt           | Specifies the input file name tag. The default is <b>upa_cvt</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of raw or grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII grid or raw file</li> <li><b>wxpb</b> - WXP binary grid file</li> <li><b>raw</b> - WXP ASCII raw file</li> <li><b>cdf</b> - NetCDF binary grid file</li> </ul> <p>Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>upa_raw</b> or <b>upa_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type:</p> <p>wxp:upa_grd</p> |

|   |                              |  |   |
|---|------------------------------|--|---|
|   |                              |  | A full filename can be specified which can contain name convention tags.  |
| <b>-pf=</b> <i>file_param</i>   | <a href="#">file_param</a>   | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul> |
| <b>-cf=</b> <i>cityfile</i>   | <a href="#">city_file</a>    | <b>upa.cty</b>   | Specifies the name of the city database file used in finding station locations.   |
| <b>-mf=</b> <i>mapfile</i> [,...]   | <a href="#">map_file</a>     | <b>wxp.map</b> or<br><b>cont.dat</b> ,<br><b>country.dat</b> for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]   | <a href="#">current</a>      | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=</b> <i>hour</i>   | <a href="#">hour</a>         | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh=</b> <i>num_hour</i>   | <a href="#">num_hour</a>     | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.   |
| <b>-id=</b> <i>identifier</i>   | <a href="#">identifier</a>   | None   | Used to plot a specific observation. If not specified, all stations will be plotted.  |
| <b>-pd=</b> <i>domain</i>   | <a href="#">plot_domain</a>  | <b>us</b>  | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-gd=</b> <i>grid_domain</i>  | <a href="#">grid_domain</a>  | Same as plot domain  | This specifies the domain to use when fitting data to a grid. If this is not specified, it is the <b>plot_domain</b> .  |
| <b>-re=</b> <i>region</i>   | <a href="#">region</a>       | Same as plot domain  | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-pr=</b> <i>prior</i>  | <a href="#">stat_prior</a>   | Depends on domain size   | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.  |
| <b>-le=</b> <i>level</i>  | <a href="#">level</a>        | User Prompt<br>Batch: <b>500</b>   | This specifies the vertical level to use in the plot. See the <b>level</b> resource for more information on what can be specified.  |
| <b>-va=</b> <i>variable</i>   | <a href="#">variable</a>     | User prompt<br>Batch: first variable listed in the <b>upairwx.var</b> file                           | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.  |
| <b>-oa=</b> <i>smooth</i><br>[, <i>radinf</i> ][, <i>passes</i> ]<br>[, <i>converg</i> ]<br>[, <i>min_sta</i> ] | <a href="#">object_param</a> | <b>1,3,1,,2,1</b>  | Specifies the objective analysis parameters. There are 5 parameters including smoothing, radius of influence, number of passes, convergence parameter and minimum number of stations.   |
| <b>-pl=</b> <i>plot_type</i>  | <a href="#">plot_type</a>    | <b>Data Plot</b><br>User prompt if plotting grids<br>Batch: fill contours or                         | Specifies the type of output plot. By default, this is a simple data plot. To produce a contour or vector plot, use the following plot types:<br><b>Contours:</b> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> </ul>  |

|  |                              |   |   |
|--|------------------------------|---|---|
|  |                              | vectors   | <ul style="list-style-type: none"> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <b>Vectors:</b> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>                | <a href="#">plot_scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>                      | <a href="#">plot_format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.  |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot_param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>              | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>                 | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.   |
| <b>-in=interval</b>                    | <a href="#">con_interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b>            | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.  |
| <b>-cb=base</b>                        | <a href="#">con_base</a>     | Lowest contour<br>value, maximum<br>wind speed                          | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.  |
| <b>-fl=font_list</b>                   | <a href="#">font_list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>                 | <a href="#">color_table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>               | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the text color for labels above and below the plot.   |
| <b>-cod=color[:attr]</b>               | <a href="#">color_data</a>   | <b>green</b>  | Specifies the color for plotted data.   |
| <b>-com=color[:attr]</b>               | <a href="#">color_map</a>    | <b>red</b>  | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>              | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.  |
| <b>-cowl=color[:attr]</b>              | <a href="#">color_wind</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for wind barbs.   |
| <b>-cocm=color[:attr]</b>              | <a href="#">color_cmap</a>   | <b>black</b>  | Specifies the color for color fill contour maps.  |
| <b>-coco=color[:attr]</b>              | <a href="#">color_cont</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for line contours.  |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color_fill</a>   | A predefined<br>color set based<br>on the <b>wxp.clr</b><br>color table | Specifies the color fill contour color set. See the <a href="#">color_fill</a> resource for more details.   |
| <b>-cola=color[:attr]</b>              | <a href="#">color_label</a>  | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the line contour label color.   |
| <b>-cocl=color[:attr]</b>              | <a href="#">color_clabel</a> | <b>black</b>  | Specifies the color fill contour label color.   |
| <b>-de=device[,...]</b>                | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                                    | Specifies the output device.  |

|   |                            |   |  |
|---|----------------------------|---|--|
| <b>-ti=title</b>                                    | <a href="#">title</a>      | Program name                                    | Specifies the window title.  |
| <b>-ic</b>  | <a href="#">icon</a>       | <b>Off</b>                                      | Specifies whether to open window in iconified mode.  |
| <b>-bg=color</b>                                    | <a href="#">background</a> | <b>black</b><br><b>white</b> for inverse plots  | Specifies the background color of the window.  |
| <b>-ge=geometry</b>                                 | <a href="#">geometry</a>   | <b>640x480</b>                                  | Specifies the window and paper geometry.   |
| <i>filename[#seq]</i><br><i>[filename[#seq]...]</i> | <a href="#">filename</a>   | None<br>User Prompt<br>Batch: <b>current=la</b> | The name of the upper air converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified to generate a loop. |

## DESCRIPTION

This program plots upper air data on a background map. The input to the program is a upper air converted file produced by the [uacvt](#) programs. Then, based on a plot domain and a variable selected by the user, a plot is generated. Overlay plots, contour plots and loops can also be generated if the appropriate options are set.

The programs starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **upa\_cvt** tag but this can be changed with the [in file](#) resource. To generate a loop, the [num hour](#) resource can be specified. This will tell the program to loop through this number of hours.

Next, the program prompts the user for the vertical level to plot. This is done through a level menu which has predefined levels and a prompt so the user can specify a level. The [level](#) resource can also be used. The levels are set up with the **upairwx.lev** file.

The domain defaults to the contiguous US unless changed through the [plot domain](#) resource. The domain can be interactively changed using the mouse once the plot has been generated.

Next, the city database is read in and a cross-reference table is setup. Since converted data is not saved with location information, the program will refer to this table to location each station in the input file.

The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the **upairwx.var** file. The internal variables are:

### Internal Variables

| Variable                                   | Description and units                  | Plots   |
|--|--|---|
| <b>city</b>                                | All stations in the data file          | string  |
| <b>temp</b>                                | Temperature (F)                        | value   |
| <b>dewp</b>                                | Dewpoint (F)                           | value   |
| <b>dewd</b>                                | Dewpoint depression (F)                | value   |
| <b>wdir</b>                                | Wind direction (deg)                   | value   |
| <b>wspd</b>                                | Wind speed (knt)                       | value   |
| <b>wgst</b>                                | Wind gust (knt)                        | value   |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)         | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)   | value   |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)   | value   |
| <b>uwndg</b>                               | U wind component - grid relative (m/s) | value   |

|                |  |       |
|----------------|--|-------|
| <b>vwndg</b>   | V wind component - grid relative (m/s)   | value |
| <b>hght</b>    | Geopotential height (m)  | value |
| <b>hag</b>     | Geopotential height above ground (m)   | value |
| <b>pres</b>    | Pressure (mb)  | value |
| <b>pag</b>     | Pressure above ground (mb)   | value |
| <b>vpres</b>   | Vapor pressure (mb)  | value |
| <b>mrat</b>    | Mixing ratio (g/kg)  | value |
| <b>shum</b>    | Specific humidity (g/kg)   | value |
| <b>rhum</b>    | Relative humidity (%)  | value |
| <b>wetblb</b>  | Wetbulb temperature (K)  | value |
| <b>thetaw</b>  | Wetbulb potential temperature (K)  | value |
| <b>thetae</b>  | Equivalent potential temperature (K)   | value |
| <b>thetav</b>  | Virtual potential temperature (K)  | value |
| <b>theta</b>   | Potential temperature (K)  | value |
| <b>vtemp</b>   | Virtual temperature (K)  | value |
| <b>thick</b>   | Thickness between two layers (m)   | value |
| <b>frz</b>     | Lowest freezing level (mb) -- height if units are km, m or ft  | value |
| <b>wb0</b>     | Wet bulb zero (mb) -- height if units are km, m or ft  | value |
| <b>ccl</b>     | Convective condensation level (mb) -- height if units are km, m or ft  | value |
| <b>convt</b>   | Convective temperature (K)   | value |
| <b>pwat</b>    | Precipitable water (in)  | value |
| <b>maxt</b>    | Estimated maximum temperature (K)  | value |
| <b>tlapse</b>  | Potential temperature lapse rate (dK/m)  | value |
| <b>tlapsep</b> | Potential temperature lapse rate over pressure (dK/mb)   | value |
| <b>lapse</b>   | Temperature lapse rate (dK/m)  | value |
| <b>ststab</b>  | Static stability ( $m^3/Pakg$ )  | value |
| <b>tlapse</b>  | Potential temperature lapse rate (dK/m)  | value |
| <b>lift</b>    | Lifted index (dC) -- parcel type dependent on level resource   | value |
| <b>cap</b>     | Cap strength (dC) -- parcel type dependent on level resource   | value |
| <b>lcl</b>     | Lifted condensation level (mb) -- parcel type dependent on level resource -- height if units are km, m or ft | value |
| <b>lfc</b>     | Level of free convection (mb) -- parcel type dependent on level resource -- height if units are km, m or ft  | value |
| <b>el</b>      | Equilibrium level (mb) -- parcel type dependent on level resource -- height if units are km, m or ft         | value |
| <b>mpl</b>     | Maximum parcel level (mb) -- parcel type dependent on level resource -- height if units are km, m or ft      | value |
| <b>cape</b>    | Convective available potential energy (J/kg) -- parcel type dependent on level resource                      | value |
| <b>cinh</b>    | Convective inhibition (J/kg) -- parcel type dependent on level resource                                      | value |
| <b>show</b>    | Showalter index (dC)   | value |
| <b>tt</b>      | Total totals index (dC)  | value |
| <b>vtot</b>    | Vertical totals index (dC)   | value |
| <b>ctot</b>    | Cross totals index (dC)  | value |

|              |   |       |
|--------------|---|-------|
| <b>ki</b>    | K index ()  | value |
| <b>sweat</b> | Sweat index ()  | value |
| <b>ei</b>    | Energy index (cal)  | value |
| <b>shear</b> | Vertical wind shear (/s) -- layer can be adjusted with level resource   | value |
| <b>pshr</b>  | Positive - Vertical wind shear (/s)   | value |
| <b>nshr</b>  | Negative - Vertical wind shear (/s)   | value |
| <b>sdir</b>  | Estimated storm direction (deg)   | value |
| <b>sspd</b>  | Estimated storm speed (m/s)   | value |
| <b>dshr</b>  | Storm relative directional shear (deg)  | value |
| <b>hel</b>   | 0 - 3000 mb AG storm relative helicity (m <sup>2</sup> /s <sup>2</sup> ) -- layer can be adjusted with level resource | value |
| <b>ehi</b>   | Energy helicity index ()  | value |
| <b>brn</b>   | Bulk Richardson number ()   | value |
| <b>bshr</b>  | Bulk shear (m/s)  | value |

These parameters can be accessed by name if they have not been redefined in the **upairwx.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

There are several derived variables from this list which are defined in the **upairwx.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable      | Description  | Plots  |
|---------------|--|--------|
| <b>all</b>    | All data, a composite plot of temperature, dewpoint, geopotential height (3 digit) and winds | comp   |
| <b>all0</b>   | Same as <b>all</b> except heights in 10 meters (useful for higher level plots)               | comp   |
| <b>allh</b>   | Same as <b>all</b> except all digits of heights are shown                                    | comp   |
| <b>alld</b>   | Same as <b>all</b> except dewpoint depression is plotted instead of dewpoint                 | comp   |
| <b>alld0</b>  | Same as <b>all0</b> except dewpoint depression is plotted instead of dewpoint                | comp   |
| <b>allp</b>   | Same as <b>all</b> except that pressure is plotted instead of height                         | comp   |
| <b>wind</b>   | A vector quantity of <uwndg,vwndg>   | vector |
| <b>vect</b>   | Wind vector plot   | vector |
| <b>strm</b>   | Streamline plot  | vector |
| <b>ugeos</b>  | U geostrophic wind component (m/s)   | grid   |
| <b>vgeos</b>  | V geostrophic wind component (m/s)   | grid   |
| <b>geos</b>   | Geostrophic winds -- vector quantity of <ugeos,vgeos>  | vector |
| <b>gspd</b>   | Geostrophic wind speed (m/s)   | grid   |
| <b>ageos</b>  | Ageostrophic winds   | vector |
| <b>avect</b>  | Ageostrophic wind vectors (m/s)  | vector |
| <b>uq</b>     | U component of the Q vector  | grid   |
| <b>vq</b>     | V component of the Q vector  | grid   |
| <b>qvect</b>  | Q vectors  | vector |
| <b>qvconv</b> | Q vector convergence   | grid   |
| <b>spres</b>  | Station pressure (mb)  | value  |
| <b>th50</b>   | 1000-500 mb thickness (m)  | value  |
| <b>mrat</b>   | Mixing ratio (g/kg) to 1 decimal place   | value  |
| <b>shum</b>   | Specific humidity (g/kg) to 1 decimal place  | value  |
| <b>conv</b>   | Wind convergence (/s) positive is convergence  | grid   |
| <b>mconv</b>  | Moisture convergence (/s)  | grid   |

|              |  |      |
|--------------|--|------|
| <b>rvort</b> | Relative vorticity (/s)                | grid |
| <b>avort</b> | Absolute vorticity (/s)                | grid |
| <b>tadv</b>  | Temperature advection (K/s)            | grid |
| <b>madv</b>  | Moisture advection (/s)                | grid |
| <b>vadv</b>  | Vorticity advection (/s <sup>2</sup> ) | grid |

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the converted file is opened and data read in. Any additional computations are performed and the data are plotted. The background map will plot first unless the output is a color fill contour in which the map plots last. If a simple plot is specified such as "**temp**", the data will be plotted to 0 decimal places on the map. Formatting of the number can be done with the **plot\_format** resource. Composite plots can either be specified with the **variable** resource (.var file, resource file, command line or variable prompt). The **variable** resource allows the user to control much of the data formatting.

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

### Contour/Vector Plots

By default, this program plots data but it can produce grids that can either be contoured or plotted as vectors or streamlines. By specifying the **plot\_type** resource to a contour or vector type, the program will generate gridded fields rather than plotting the data. The gridding process uses an objective analysis scheme to produce the grids. The parameters for this process are specified in the **object\_param** resource. Once the grids have been created, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display in much the same fashion as for the data plot.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.
- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

### Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

### EXAMPLES

```
upairwx -cu=la -le=850 -va=all -de=d
```

This plots a 850 mb composite plot containing temperature, dewpoint, geopotential height and winds for the contiguous US.

```
upairwx -cu=la -le=850-500mb -va=temp -de=d
```

This plots the mean temperature for the layer of 850 to 500 mb.

```
upairwx -cu=la -le=320k -va=allp -de=d
```

This plots a composite plot for the 320 K isentropic surface. In this case, the composite plot includes the pressure, not the height.

```
upairwx -cu=la -le=700mb -va=temp -of=raw -pl=none
```

This will generate raw file based on the **upa\_raw** name convention. A specific filename can be used instead of **raw**.

```
upairwx -cu=la -le=100m_ag -va=temp -pl=cf -in=2 -de=d
```

This will produce a contour plot of 100 meter above ground temperatures. Without the "-pl=cf", the program will just plot the temperature data.

## FILES

- **upa.cty** - the surface station location database file
- **upairwx.lev** - the level menu
- **upairwx.var** - the variable menu and definitions

## SEE ALSO

- [uacvt](#) - the upper air decoding program
- [upcalc](#) - the upper air contouring program
- [uacalplt](#) - the sounding display program

# UPCALC

## NAME

**upcalc** - Upper air contouring program

## SYNOPSIS

**upcalc** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description   |
|------------------------------|----------------------------|-------------------|---|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.   |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxpdef           | Sets the name of the resource file.   |
| <b>-na=name</b>              | <a href="#">name</a>       | upcalc            | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode   |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> <li>gridding information - <b>out3</b></li> </ul>  |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.   |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.  |
| <b>-rp=rawpath</b>           | <a href="#">raw_path</a>   | current directory | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.   |
| <b>-gp=gridpath</b>          | <a href="#">grid_path</a>  | current directory | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.   |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.   |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | upa_cvt           | Specifies the input file name tag. The default is <b>upa_cvt</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.  |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of raw or grid file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li><b>wxp</b> - WXP ASCII grid or raw file</li> <li><b>wxpb</b> - WXP binary grid file</li> <li><b>raw</b> - WXP ASCII raw file</li> <li><b>cdf</b> - NetCDF binary grid file</li> </ul> <p>Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>upa_raw</b> or <b>upa_grd</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type:</p> <p>wxp:upa_grd</p> |

|   |                              |  |   |
|---|------------------------------|--|---|
|   |                              |  | A full filename can be specified which can contain name convention tags.  |
| <b>-pf=</b> <i>file_param</i>   | <a href="#">file_param</a>   | User prompt<br>Batch: <b>use</b>   | This specifies how to treat output files that exist: <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>App</b> - append the information onto the end of the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul> |
| <b>-cf=</b> <i>cityfile</i>   | <a href="#">city_file</a>    | <b>upa.cty</b>   | Specifies the name of the city database file used in finding station locations.   |
| <b>-mf=</b> <i>mapfile</i> [,...]   | <a href="#">map_file</a>     | <b>wxp.map</b> or<br><b>cont.dat</b> ,<br><b>country.dat</b> for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.  |
| <b>-cu=</b> [ <i>hour</i>   <b>la</b> ]   | <a href="#">current</a>      | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=</b> <i>hour</i>   | <a href="#">hour</a>         | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh=</b> <i>num_hour</i>   | <a href="#">num_hour</a>     | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.   |
| <b>-id=</b> <i>identifier</i>   | <a href="#">identifier</a>   | None   | Used to plot a specific observation. If not specified, all stations will be plotted.  |
| <b>-pd=</b> <i>domain</i>   | <a href="#">plot_domain</a>  | User prompt<br>Batch: first<br>domain in<br><b>wxp.reg</b> file.                                     | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-gd=</b> <i>grid_domain</i>  | <a href="#">grid_domain</a>  | Same as plot<br>domain   | This specifies the domain to use when fitting data to a grid. If this is not specified, it is the <b>plot_domain</b> .  |
| <b>-re=</b> <i>region</i>   | <a href="#">region</a>       | Same as plot<br>domain   | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-pr=</b> <i>prior</i>  | <a href="#">stat_prior</a>   | Depends on<br>domain size  | This specifies the lowest priority to use in the plot. Priorities range from 1 to 7.  |
| <b>-le=</b> <i>level</i>  | <a href="#">level</a>        | User Prompt<br>Batch: <b>500</b>   | This specifies the vertical level to use in the plot. See the <b>level</b> resource for more information on what can be specified.  |
| <b>-va=</b> <i>variable</i>   | <a href="#">variable</a>     | User prompt<br>Batch: first<br>variable listed in<br>the <b>upcalc.var</b><br>file                   | This specifies the variable to plot along with its formatting attributes. Multiple variables can be specified separated by commas for overlay plots. A list of valid variables is given below.  |
| <b>-oa=</b> <i>smooth</i><br>[, <i>radinf</i> ][, <i>passes</i> ]<br>[, <i>converg</i> ]<br>[, <i>min_sta</i> ] | <a href="#">object_param</a> | <b>1,3,1,2,1</b>   | Specifies the objective analysis parameters. There are 5 parameters including smoothing, radius of influence, number of passes, convergence parameter and minimum number of stations.   |
| <b>-pl=</b> <i>plot_type</i>  | <a href="#">plot_type</a>    | User prompt<br>Batch: fill<br>contours or  | Specifies the type of output plot. By default, the user is prompted for the plot type. To produce a contour or vector plot, use the following plot types:   |

|  |                              |   |  |
|--|------------------------------|---|--|
|  |                              | vectors   | <p><b>Contours:</b></p> <ul style="list-style-type: none"> <li>• <b>cf</b> -- color fill contours</li> <li>• <b>ln</b> -- line contours</li> <li>• <b>dln</b> -- dashed line contours</li> <li>• <b>ndln</b> -- negative dash line contours</li> <li>• <b>lcf</b> -- both line and color fill contours</li> <li>• <b>gvalue</b> -- plot gridpoint values</li> </ul> <p><b>Vectors:</b></p> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>                | <a href="#">plot scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm=format</b>                      | <a href="#">plot format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp=param</b><br>[,param...]        | <a href="#">plot param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot param</a> resource for more details.  |
| <b>-dr=drawable[,...]</b>              | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.   |
| <b>-la=label[,...]</b>                 | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-in=interval</b>                    | <a href="#">con interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b>            | Specifies the contour interval to use in contouring grids. This can also be used to color code plotted data and vectors.   |
| <b>-cb=base</b>                        | <a href="#">con base</a>     | Lowest contour<br>value, maximum<br>wind speed                          | Specifies the base value for color fill contours, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.   |
| <b>-fl=font_list</b>                   | <a href="#">font list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.   |
| <b>-ct=color_table</b>                 | <a href="#">color table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |
| <b>-cot=color[:attr]</b>               | <a href="#">color text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the text color for labels above and below the plot.  |
| <b>-cod=color[:attr]</b>               | <a href="#">color data</a>   | <b>green</b>  | Specifies the color for plotted data.  |
| <b>-com=color[:attr]</b>               | <a href="#">color map</a>    | <b>red</b>  | Specifies the color for background maps.   |
| <b>-coln=color[:attr]</b>              | <a href="#">color line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.   |
| <b>-cowl=color[:attr]</b>              | <a href="#">color wind</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for wind barbs.  |
| <b>-cocm=color[:attr]</b>              | <a href="#">color cmap</a>   | <b>black</b>  | Specifies the color for color fill contour maps.   |
| <b>-coco=color[:attr]</b>              | <a href="#">color cont</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the color for line contours.   |
| <b>-cof=[val:]color</b><br>[:attr],... | <a href="#">color fill</a>   | A predefined<br>color set based<br>on the <b>wxp.clr</b><br>color table | Specifies the color fill contour color set. See the <a href="#">color_fill</a> resource for more details.  |
| <b>-cola=color[:attr]</b>              | <a href="#">color label</a>  | <b>white</b><br><b>black</b> for inverse                                | Specifies the line contour label color.  |

|   |                             |  |  |
|---|-----------------------------|--|--|
|   |                             | plots.   |  |
| <b>-cocl=</b> <i>color[:attr]</i>                   | <a href="#">color_label</a> | <b>black</b>                                   | Specifies the color fill contour label color.  |
| <b>-de=</b> <i>device[,...]</i>                     | <a href="#">device</a>      | User prompt<br>Batch: <b>display</b>           | Specifies the output device.   |
| <b>-ti=</b> <i>title</i>                            | <a href="#">title</a>       | Program name                                   | Specifies the window title.  |
| <b>-ic</b>  | <a href="#">icon</a>        | <b>Off</b>                                     | Specifies whether to open window in iconified mode.  |
| <b>-bg=</b> <i>color</i>                            | <a href="#">background</a>  | <b>black</b><br><b>white</b> for inverse plots | Specifies the background color of the window.  |
| <b>-ge=</b> <i>geometry</i>                         | <a href="#">geometry</a>    | <b>640x480</b>                                 | Specifies the window and paper geometry.   |
| <i>filename[#seq]</i><br><i>[filename[#seq]...]</i> | <a href="#">filename</a>    | None<br>User Pompt<br>Batch: <b>current=la</b> | The name of the upper air converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files. More than one file can be specified to generate a loop. |

## DESCRIPTION

This program contours upper air data on a background map. The input to the program is a upper air converted file produced by the [uacvt](#) programs. Then, based on a plot domain and a variable selected by the user, a plot is generated. Overlay plots, plots and loops can also be generated if the appropriate options are set.

The programs starts off by prompting the user for input converted file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the **upa\_cvt** tag but this can be changed with the [in file](#) resource. To generate a loop, the [num hour](#) resource can be specified. This will tell the program to loop through this number of hours.

Next, the program prompts the user for the vertical level to plot. This is done through a level menu which has predefined levels and a prompt so the user can specify a level. The [level](#) resource can also be used. The levels are set up with the **upairwx.lev** file.

The domain defaults to the contiguous US unless changed through the [plot domain](#) resource. The domain can be interactively changed using the mouse once the plot has been generated.

Next, the city database is read in and a cross-reference table is setup. Since converted data is not saved with location information, the program will refer to this table to location each station in the input file.

The program will now prompt the user for the particular output variable to plot. The program will show a menu of commonly used variables but many more are defined internally by the program and in the **upcalc.var** file. The internal variables are:

### Internal Variables

| Variable                                   | Description and units                  | Plots   |
|--|--|---|
| <b>temp</b>                                | Temperature (F)                        | value   |
| <b>dewp</b>                                | Dewpoint (F)                           | value   |
| <b>dewd</b>                                | Dewpoint depression (F)                | value   |
| <b>wdir</b>                                | Wind direction (deg)                   | value   |
| <b>wspd</b>                                | Wind speed (knt)                       | value   |
| <b>wgst</b>                                | Wind gust (knt)                        | value   |
| <b>wbrbc</b><br><b>wbrb</b><br><b>wind</b> | Wind speed and direction (knt)         | value <sub>0</sub> =dir,<br>value <sub>1</sub> =spd |
| <b>uwnd</b>                                | U wind component - EW relative (m/s)   | value   |
| <b>vwnd</b>                                | V wind component - NS relative (m/s)   | value   |
| <b>uwndg</b>                               | U wind component - grid relative (m/s) | value   |
| <b>vwndg</b>                               | V wind component - grid relative (m/s) | value   |

|                |  |       |
|----------------|--|-------|
| <b>hght</b>    | Geopotential height (m)  | value |
| <b>hag</b>     | Geopotential height above ground (m)   | value |
| <b>pres</b>    | Pressure (mb)  | value |
| <b>pag</b>     | Pressure above ground (mb)   | value |
| <b>vpres</b>   | Vapor pressure (mb)  | value |
| <b>mrat</b>    | Mixing ratio (g/kg)  | value |
| <b>shum</b>    | Specific humidity (g/kg)   | value |
| <b>rhum</b>    | Relative humidity (%)  | value |
| <b>wetblb</b>  | Wetbulb temperature (K)  | value |
| <b>thetaw</b>  | Wetbulb potential temperature (K)  | value |
| <b>thetae</b>  | Equivalent potential temperature (K)   | value |
| <b>thetav</b>  | Virtual potential temperature (K)  | value |
| <b>theta</b>   | Potential temperature (K)  | value |
| <b>vtemp</b>   | Virtual temperature (K)  | value |
| <b>thick</b>   | Thickness between two layers (m)   | value |
| <b>frz</b>     | Lowest freezing level (mb) -- height if units are km, m or ft  | value |
| <b>wb0</b>     | Wet bulb zero (mb) -- height if units are km, m or ft  | value |
| <b>ccl</b>     | Convective condensation level (mb) -- height if units are km, m or ft  | value |
| <b>convt</b>   | Convective temperature (K)   | value |
| <b>pwat</b>    | Precipitable water (in)  | value |
| <b>maxt</b>    | Estimated maximum temperature (K)  | value |
| <b>tlapse</b>  | Potential temperature lapse rate (dK/m)  | value |
| <b>tlapsep</b> | Potential temperature lapse rate over pressure (dK/mb)   | value |
| <b>lapse</b>   | Temperature lapse rate (dK/m)  | value |
| <b>ststab</b>  | Static stability ( $m^3/Pakg$ )  | value |
| <b>tlapse</b>  | Potential temperature lapse rate (dK/m)  | value |
| <b>lift</b>    | Lifted index (dC) -- parcel type dependent on level resource   | value |
| <b>cap</b>     | Cap strength (dC) -- parcel type dependent on level resource   | value |
| <b>lcl</b>     | Lifted condensation level (mb) -- parcel type dependent on level resource -- height if units are km, m or ft | value |
| <b>lfc</b>     | Level of free convection (mb) -- parcel type dependent on level resource -- height if units are km, m or ft  | value |
| <b>el</b>      | Equilibrium level (mb) -- parcel type dependent on level resource -- height if units are km, m or ft         | value |
| <b>mpl</b>     | Maximum parcel level (mb) -- parcel type dependent on level resource -- height if units are km, m or ft      | value |
| <b>cape</b>    | Convective available potential energy (J/kg) -- parcel type dependent on level resource                      | value |
| <b>cinh</b>    | Convective inhibition (J/kg) -- parcel type dependent on level resource                                      | value |
| <b>show</b>    | Showalter index (dC)   | value |
| <b>tt</b>      | Total totals index (dC)  | value |
| <b>vtot</b>    | Vertical totals index (dC)   | value |
| <b>ctot</b>    | Cross totals index (dC)  | value |
| <b>ki</b>      | K index ( )  | value |
| <b>sweat</b>   | Sweat index ( )  | value |
| <b>ei</b>      | Energy index (cal)   | value |
| <b>shear</b>   | Vertical wind shear (/s) -- layer can be adjusted with level resource  | value |
| <b>pshr</b>    | Positive - Vertical wind shear (/s)  | value |
| <b>nshr</b>    | Negative - Vertical wind shear (/s)  | value |
| <b>sdir</b>    | Estimated storm direction (deg)  | value |
| <b>sspd</b>    | Estimated storm speed (m/s)  | value |
| <b>dshr</b>    | Storm relative directional shear (deg)   | value |

|             |   |       |
|-------------|---|-------|
| <b>hel</b>  | 0 - 3000 mb AG storm relative helicity ( $m^2/s^2$ ) -- layer can be adjusted with level resource | value |
| <b>ehi</b>  | Energy helicity index ( )   | value |
| <b>brn</b>  | Bulk Richardson number ( )  | value |
| <b>bshr</b> | Bulk shear (m/s)  | value |

These parameters can be accessed by name if they have not been redefined in the **upcalc.var** file. Otherwise, prepending a "+" to the variable name will guarantee these variables will appear.

There are several derived variables from this list which are defined in the **upcalc.var** file. These include composite plots, overlay plots and computed grids.

#### Derived/Composite Variables

| Variable      | Description   | Plots  |
|---------------|---|--------|
| <b>wind</b>   | A vector quantity of <uwndg,vwndg>                    | vector |
| <b>vect</b>   | Wind vector plot                                      | vector |
| <b>strm</b>   | Streamline plot                                       | vector |
| <b>ugeos</b>  | U geostrophic wind component (m/s)                    | grid   |
| <b>vgeos</b>  | V geostrophic wind component (m/s)                    | grid   |
| <b>geos</b>   | Geostrophic winds -- vector quantity of <ugeos,vgeos> | vector |
| <b>gspd</b>   | Geostrophic wind speed (m/s)                          | grid   |
| <b>ageos</b>  | Ageostrophic winds -- vector quantity                 | vector |
| <b>avect</b>  | Ageostrophic wind vectors (m/s)                       | vector |
| <b>uq</b>     | U component of the Q vector                           | grid   |
| <b>vq</b>     | V component of the Q vector                           | grid   |
| <b>qvect</b>  | Q vectors -- vector quantity                          | vector |
| <b>qvconv</b> | Q vector convergence                                  | grid   |
| <b>spres</b>  | Station pressure (mb)                                 | value  |
| <b>th50</b>   | 1000-500 mb thickness (m)                             | value  |
| <b>mrat</b>   | Mixing ratio (g/kg) to 1 decimal place                | value  |
| <b>shum</b>   | Specific humidity (g/kg) to 1 decimal place           | value  |
| <b>conv</b>   | Wind convergence (/s) positive is convergence         | grid   |
| <b>mconv</b>  | Moisture convergence (/s)                             | grid   |
| <b>rvort</b>  | Relative vorticity (/s)                               | grid   |
| <b>avort</b>  | Absolute vorticity (/s)                               | grid   |
| <b>tadv</b>   | Temperature advection (K/s)                           | grid   |
| <b>madv</b>   | Moisture advection (/s)                               | grid   |
| <b>vadv</b>   | Vorticity advection (/s <sup>2</sup> )                | grid   |

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the converted file is opened and data read in. Any additional computations are performed to produce a derived variable and fit the data to a grid. The background map will plot first unless the output is a color fill contour in which the map plots last. The gridding process uses an objective analysis scheme to produce the grids. The parameters for this process are specified in the **object\_param** resource. Once the grids have been created, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display.

There are several plot types. For contour plots, there is:

- **ln** - line contours
- **dln** - dashed line contours
- **ndln** - negative dashed line contours. All contours less than 0 are dashed. The breakoff can be reset with the **con\_base** resource.

- **cf** - color fill contours. These are contours where the areas between the contour lines are filled with a progressive color scheme. By default, this is a 15 color scale in which the colors wrap around if there are more than 15 color level. The color scheme can be set by the **color\_fill** resource.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed. When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## Output Files

By default, this program does not produce output files. If a raw or grid file output is desired, use the **out\_file** resource to enable this. By default, this will create a WXP ASCII raw or grid file depending on the plot type. The **out\_file** can be a filename, a file tag (lookup in file name convention file) or a name convention (with wildcards). The output can be GIF file by specifying **gif** as the file type. It may be desirable to turn plotting of when producing output files. To do this, specify **none** for the **plot\_type** resource.

## EXAMPLES

```
upcalc -cu=la -le=500 -va=temp -pl=cf -in=2 -de=d
```

This produces a color fill contour plot of 500 mb temperature for the contiguous US region at an interval of 2.

```
upcalc -cu=la -le=850 -va=hght -of=wxp -pl=none
```

This will generate grid file based on the **upa\_grd** name convention. A specific filename can be used instead of **wxp**.

## FILES

- **upa.cty** - the upper air station location database file
- **upcalc.lev** - level menu definition
- **upcalc.var** - variable menu and definitions

## SEE ALSO

- [uacvt](#) - the upper air data decoding program
- [upairwx](#) - the upper air data plotting program
- [uacalplt](#) - the upper air sounding display program

# VECTOR

## NAME

**vector** - Vector display program

## SYNOPSIS

**vector** [*parameters...*] *filename*

## PARAMETERS

| Command Line             | Resource                    | Default  | Description  |
|--------------------------|-----------------------------|--|--|
| <b>-h</b>                | <a href="#">help</a>        | No   | Lists basic help information.  |
| <b>-df=filename</b>      | <a href="#">default</a>     | .wxpdef  | Sets the name of the resource file.  |
| <b>-na=name</b>          | <a href="#">name</a>        | <b>vector</b>  | Specifies the name used in resource file parsing.  |
| <b>-ba</b>               | <a href="#">batch</a>       | No   | Run program in batch mode  |
| <b>-me=level</b>         | <a href="#">message</a>     | <b>out2</b>  | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>file information - <b>mess</b></li> </ul>   |
| <b>-fp=filepath</b>      | <a href="#">file_path</a>   | current directory  | Specifies location of database files.  |
| <b>-dp=datapath</b>      | <a href="#">data_path</a>   | current directory  | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.   |
| <b>-cp=conpath</b>       | <a href="#">con_path</a>    | current directory  | Specifies the location (path) of the input surface converted data files. This may be modified in the name convention file.   |
| <b>-rp=rawpath</b>       | <a href="#">raw_path</a>    | current directory  | Specifies the location of the output raw files generated by the program. These files will not be generated unless the <b>out_file</b> resource is specified and the output is plotted data.  |
| <b>-gp=gridpath</b>      | <a href="#">grid_path</a>   | current directory  | Specifies the location of the output grid files generated by the program. These files are not generated unless the <b>out_file</b> resource is specified and the output is a gridded field.  |
| <b>-ip=imagepath</b>     | <a href="#">image_path</a>  | current directory  | Specifies the default location of the output GIF images generated by the program.  |
| <b>-nc=name_conv</b>     | <a href="#">name_conv</a>   | <b>name_conv</b>   | The name convention file specifies how files are named in WXP. This sets which name convention file to use.  |
| <b>-if=in_file</b>       | <a href="#">in_file</a>     | <b>grd</b>   | Specifies the input file name tag. The default is <b>grd</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well.   |
| <b>-mf=mapfile[,...]</b> | <a href="#">map_file</a>    | <b>wxp.map</b> or<br><b>cont.dat</b> ,<br><b>country.dat</b> for<br>domains outside<br>North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu=[hour]la</b>      | <a href="#">current</a>     | None   | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file. |
| <b>-ho=hour</b>          | <a href="#">hour</a>        | None   | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh=num_hour</b>      | <a href="#">num_hour</a>    | <b>0</b> (no loop)   | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.  |
| <b>-pd=domain</b>        | <a href="#">plot_domain</a> | User prompt<br>Batch: first  | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |

|  |                              |   |   |
|--|------------------------------|---|---|
|  |                              | domain in <b>wxp.reg</b> file.  |   |
| <b>-re=region</b>  | <a href="#">region</a>       | Same as plot domain   | This resource is synonymous with the <b>plot_domain</b> .   |
| <b>-pl=plot_type</b>   | <a href="#">plot_type</a>    | User prompt<br>Batch: fill<br>vectors or<br>vectors                     | Specifies the type of output plot. By default, the user is prompted for the plot type. To produce a vector or vector plot, use the following plot types:<br><b>Vectors:</b> <ul style="list-style-type: none"> <li>• <b>vect</b> -- vectors/arrows</li> <li>• <b>strm</b> -- streamlines</li> <li>• <b>barb</b> -- wind barbs</li> <li>• <b>none</b> -- for no output plot</li> </ul> |
| <b>-sc=scale_factor</b>  | <a href="#">plot_scale</a>   | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.   |
| <b>-fm=format</b>  | <a href="#">plot_format</a>  | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.  |
| <b>-pp=param</b><br>[,param...]  | <a href="#">plot_param</a>   | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-dr=drawable[,...]</b>  | <a href="#">draw</a>         | <b>all</b>  | Specifies which part of the plot is to be drawn.  |
| <b>-la=label[,...]</b>   | <a href="#">label</a>        | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.   |
| <b>-in=interval</b>  | <a href="#">con_interval</a> | None for plots,<br>User prompt for<br>grids. Batch: <b>0</b>            | Specifies the interval to use in coloring vectors.  |
| <b>-cb=base</b>  | <a href="#">con_base</a>     | Maximum wind<br>speed   | Specifies the base value for color fill vectors, and wind speed per grid distance. For more details, see the <a href="#">con_base</a> resource.   |
| <b>-fl=font_list</b>   | <a href="#">font_list</a>    | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>   | <a href="#">color_table</a>  | <b>wxp.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-cot=color[:attr]</b>   | <a href="#">color_text</a>   | <b>white</b><br><b>black</b> for inverse<br>plots.                      | Specifies the text color for labels above and below the plot.   |
| <b>-com=color[:attr]</b>   | <a href="#">color_map</a>    | <b>red</b>  | Specifies the color for background maps.  |
| <b>-coln=color[:attr]</b>  | <a href="#">color_line</a>   | <b>magenta:st=dsh</b>   | Specifies the color for lat/lon lines.  |
| <b>-coco=color[:attr]</b>  | <a href="#">color_data</a>   | <b>white</b>  | Specifies the color for vectors.  |
| <b>-cof=[val:]color</b><br>[:attr],...   | <a href="#">color_fill</a>   | A predefined<br>color set based<br>on the <b>wxp.clr</b><br>color table | Specifies the color fill vector color set. See the <a href="#">color_fill</a> resource for more details.  |
| <b>-de=device[,...]</b>  | <a href="#">device</a>       | User prompt<br>Batch: <b>display</b>                                    | Specifies the output device.  |
| <b>-ti=title</b>   | <a href="#">title</a>        | Program name  | Specifies the window title.   |
| <b>-ic</b>   | <a href="#">icon</a>         | <b>Off</b>  | Specifies whether to open window in iconified mode.   |
| <b>-bg=color</b>   | <a href="#">background</a>   | <b>black</b><br><b>white</b> for inverse<br>plots                       | Specifies the background color of the window.   |
| <b>-ge=geometry</b>  | <a href="#">geometry</a>     | <b>640x480</b>  | Specifies the window and paper geometry.  |
| <b>Ufilename[+time:</b><br><b>level:var[:units]]</b><br><b>vfilename[+time:</b><br><b>level:var[:units]]</b> | <a href="#">filename</a>     | None<br>User Prompt<br>Batch: current= <b>la</b>                        | The name of the surface converted data file to be plotted. An optional sequence number can be added to designate the time for non-WXP files.  |

## DESCRIPTION

This program plots vectors for a set of two existing grid files either created by WXP or preformatted GRIB files. WXP grid files can be created with the **out\_file** resource being set on the contouring programs such as **sfccalc** and **upcalc**. For multigrid files, additional information can be tacked onto the filename to denote which grid to use. If not specified on the command line, the filename will be prompted for. The filename syntax is:

```
filename[+time:level:var[:units]]  
or  
filename[#num[:units]]
```

where **filename** is the name of the grid file to use. If a relative path is used, the value of the **grid\_path** is prepended to the filename. The filename can be "std" where standard input is used.

The extra information following the "+" or "#" defines which grid to use within a multigrid file. The value of **num** represents the sequence number of the grid in the file. Grids are numbered starting at 0 (zero). This number can be determined through a **griblook** listing.

The values of **time**, **level**, and **var** make it possible to specify the grid based on typical grid information. Possible values for each are:

- **time:** init,06,12,18,24,30,36,42,48,60,72,d3,d4,d5,d6,d7,d8,d9,d10
- **level:** sfc, sl (sea level), trop, wind, 1000, 850, 700, 500, 400, 300, 200, 100
- **var:** temp, rhum, pres, vvel, uwnd, vwnd, prec, lift

The **unit** specification allows the user to change the units of the grid before performing the math.

The user may also specify the input file using the **in\_file** resource to specify a name convention tag and the **current** resource. To generate a loop, the **num\_hour** resource can be specified. This will tell the program to loop through this number of hours.

The domain defaults to the domain of the first grid unless changed through the **plot\_domain** resource. The domain can be interactively changed using the mouse once the plot has been generated.

The program will now open the graphics window and start processing the data. The user will be prompted for the output device. Then the grid file is opened and data read in. The background map will plot first unless the output is a color fill contour in which the map plots last. Next, the maximum and minimum of the grid are displayed for the user to pick an appropriate contour interval. Then, the contour or vector plot will display.

For vectors:

- **vect** -- vectors/arrows
- **strm** -- streamlines
- **barb** -- wind barbs

When the plot is complete, output can be saved to a GIF image file, spooled to the printer or the plot can be zoomed.

When zooming the image, the plot will redraw to the new domain. It will also add more stations to the plot depending on the size of the domain. When the user is finished with the plot, the enter key can be hit which closes the window and returns the user to the variable prompt.

## EXAMPLES

```
vector -pl=vect -in=2 -de=d uwnd.grd vwnd.grd
```

This produces a wind vector plot with vectors colored at an interval of 2.

## FILES

### SEE ALSO

- [sfccalc](#) - the surface contouring program
- [upcalc](#) - the upper air contouring program
- [focalc](#) - the MOS contouring program

# WCHILL

## NAME

**wchill** - Wind chill computation program

## SYNOPSIS

**wchill** [*parameters*] [*temp speed*]

## PARAMETERS

| Command Line         | Resource                  | Default     | Description  |
|----------------------|---------------------------|-------------|--|
| <b>-h</b>            | <a href="#">help</a>      | No          | Lists basic help information.  |
| <b>-df=filename</b>  | <a href="#">default</a>   | .wxpdef     | Sets the name of the resource file.  |
| <b>-na=name</b>      | <a href="#">name</a>      | wchill      | Specifies the name used in resource file parsing.  |
| <b>-ba</b>           | <a href="#">batch</a>     | No          | Run program in batch mode  |
| <b>-me=level</b>     | <a href="#">message</a>   | out2        | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• output values - <b>print</b></li> <li>• value labels - <b>mess</b></li> <li>• table -<b>out1</b></li> </ul> |
| <b>-ta</b>           | <a href="#">table</a>     | none        | Specifies to print out a table. This is the default if temperature and speed aren't prespecified.  |
| <b>-pa=parameter</b> | <a href="#">parameter</a> | none        | Specifies additional parameters: <ul style="list-style-type: none"> <li>• <b>wcval</b> - wind chill value (cals)</li> <li>• <b>wctemp</b> - wind chill temp (F)</li> </ul>                       |
| <i>temp</i>          | <a href="#">value</a>     | none        | Specifies the temperature in F   |
| <i>speed</i>         |                           | User prompt | Specifies the wind speed in knts   |

## DESCRIPTION

The wind chill program calculates the wind chill factor from a specified temperature and wind speed. If *temp* and *wind* are not specified, the program displays a wind chill chart for reference and then prompts the user for a temperature value in F and an associated wind speed in knots. From this input data, the wind chill factor is calculated. The wind chill factor is relatively meaningless for a temperature greater than 50 F because the wind cools the skin's surface less for warmer temperatures. Also, the wind chill factor changes little for wind speeds greater than 40 knots because at that speed, the wind removes nearly all heat produced at the skin's surface. For example:

```
wchill 40 20
16.2
```

or to print a wind chill table:

```
wchill -table
Wind Chill
      Temperature(F)
      40  35  30  25  20  15  10  5  0  -5  -10  -15  -20  -25  -30
Wind(knots)
5:  36  30  25  19  14  8  3  -2  -8  -13  -19  -24  -30  -35  -40
10:  26  20  13  7  1  -6  -12  -18  -25  -31  -37  -44  -50  -56  -63
15:  20  13  6  -1  -7  -14  -21  -28  -35  -42  -49  -56  -63  -70  -77
20:  16  9  2  -6  -13  -20  -28  -35  -42  -50  -57  -64  -72  -79  -86
25:  13  6  -2  -9  -17  -25  -32  -40  -47  -55  -63  -70  -78  -85  -93
30:  11  4  -4  -12  -20  -28  -35  -43  -51  -59  -66  -74  -82  -90  -98
35:  10  2  -6  -14  -22  -30  -37  -45  -53  -61  -69  -77  -85  -93  -101
40:  9  1  -7  -15  -23  -31  -39  -47  -55  -63  -71  -79  -87  -95  -103
```

**FILES**

**SEE ALSO**

# WXP

## NAME

**wxp** - The WXP shell

## SYNOPSIS

**wxp** [*parameters*] [*command*]

## PARAMETERS

| Command Line                | Resource                  | Default     | Description                                       |
|-----------------------------|---------------------------|-------------|---|
| <b>-h</b>                   | <a href="#">help</a>      | No          | Lists basic help information.                     |
| <b>-df=<i>filename</i></b>  | <a href="#">default</a>   | .wxddef     | Sets the name of the resource file.               |
| <b>-na=<i>name</i></b>      | <a href="#">name</a>      | wxp         | Specifies the name used in resource file parsing. |
| <b>-ba</b>                  | <a href="#">batch</a>     | No          | Run program in batch mode                         |
| <b>-me=<i>level</i></b>     | <a href="#">message</a>   | out2        | Specifies level of messages to be displayed.      |
| <b>-pa=<i>parameter</i></b> | <a href="#">parameter</a> | none        | Specifies additional parameters.                  |
| <i>command</i>              | <a href="#">command</a>   | none        | Specifies a command to run.                       |
|                             |                           | User prompt |   |

## DESCRIPTION

The WXP shell provides a simple interactive environment for access to the various WXP programs via a set of nested menus. The shell initializes several parameters including the initial environment and menu setup. It then runs any commands listed on the command line. Command line commands are run and then the shell exits. Otherwise, the shell will prompt the user for the programs to run by listing a menu of commands or submenus. The shell prompt lists the current menu level name as part of its prompt. For example, the initial prompt will read "**WXP-main**>":

```
WXP: The Weather Processor - version 5.0-X11
  Copyright: Purdue Research Foundation
    Purdue University, 1989-1996
```

```
  Main Menu
```

```
  1: Parsing Programs
  2: Plotting Programs
  3: Contouring Programs
  4: Meteorological Calculations
-----
  0: Return to previous menu
 -1: Exit WXP shell
```

```
WXP-main>
```

If the user selects the plotting submenu by hitting "2", the following menu and prompt will appear:

```
  Plot Data Menu
```

```
  1: Plot Surface Data
  2: Plot Surface Meteograms
  3: Plot Upper Air Data
  4: Plot Soundings
  5: Plot ETA Model Soundings
```

```

6: Plot MDR Radar Data
7: Plot NOWRad Data
8: Plot Lightning Data
9: Plot MOS Data
10: Plot MOS Meteograms
11: Plot Maps
-----
0: Return to previous menu
-1: Exit WXP shell

WXP-plot>

```

A list of menu items will be displayed after every user command unless the user turns the menu listing off with the "**menu off**" command. By making a selection from the submenu, a WXP program will be run. For example, selecting "**1**" will print "Running: sfcwx" and then run the program:

```

Running: sfcwx
      SURFACE DATA PLOTTING (Ver 5.0-X11)

List of available files:
/usr/local/data/convert/96110800.cvt  /usr/local/data/convert/96110801.cvt
...
/usr/local/data/convert/96111120.cvt  /usr/local/data/convert/96111121.cvt
/usr/local/data/convert/96111122.cvt  /usr/local/data/convert/96111123.cvt

Enter the surface filename:

```

The WXP program will have a set of menus that the user can select from. These menu prompts will not start with "**WXP-**". If the user exists the program, control is return to the WXP shell and a shell prompt will display. You can run other programs or change menus. By convention, the menu item "**0**" goes back one menu level and "**-1**" exits the shell.

The user can supply command line parameters to the WXP programs by listing them after the menu item number. For example, "**1 -cu -me=out4**". In other words, the shell converts the "**1**" to the string "**sfcwx**" and anything else specified is passed along to "sfcwx". For example:

```
1 -cu=la -re=mw -va=all -de=d
```

At any time, you can run any existing command on the system. In other words, the WXP shell behaves like a simplified shell like C-shell. For example, you can run "**ls -las**" or invoke editors such as "vi" directly from the WXP shell. The shell first checks to see if the command you entered starts with a number (hence a menu selection) or is one of the internal shell commands like "**set**". If it isn't one of these, it attempts to run the command as listed. The shell also allows the user to set environment variables and to run simple shell scripts.

## Shell Commands

The following provides a short list of WXP shell commands that can be used to help the user manipulate various program parameters.

|                            |  |
|----------------------------|--|
| <b>back</b>                | Goes up one menu level   |
| <b>case</b>                | Lists all available cases  |
| <b>case name</b>           | Enables a specific case. Parameters are listed in the case.lup file.   |
| <b>exit</b> or <b>quit</b> | Exits the WXP shell.   |
| <b>help</b>                | This command displays a brief explanation of each WXP shell command.   |
| <b>menu</b>                | This command prints the currently selected menu.   |
| <b>menu on</b>             | Turns on the automatic listing of the menu choices. When menu listing is on, the current menu will be printed after every command completes execution. |

|                         |   |
|-------------------------|---|
| <b>menu off</b>         | Turns off the automatic listing of menus. When menu listing is off, the current menu will be printed only if the "menu" command is executed. The shell prompt lists the current menu name in order for the user to easily determine what menu has been selected.  |
| <b>set var value</b>    | This shell command sets the value of an environment variable where <i>var</i> is the name of the variable to be set and <i>value</i> is the new value for that variable. If the variable does not exist, it will be created and its value set to <i>value</i> .   |
| <b>set or show</b>      | This command lists the current settings of established environment variables. This will list the variable name followed by an equal sign and then list the value for that variable. For example:<br><br>wxpfile=/atms/bin<br><br>where <b>wxpfile</b> is the name of the environment variable and its value is <b>/atms/bin</b> .   |
| <b>uncase</b>           | Unsets case study, resets the shell to using default data.  |
| <b>unset variable</b>   | This command removes an environment variable where <i>variable</i> is the name of the variable to be deleted.   |
| <i>nn</i>               | The menus and programs may be selected by typing in the appropriate item number <i>nn</i> listed in the selected menu at the prompt. The menu item <b>0</b> always returns to the previous menu level. The menu item <b>-1</b> exits the WXP shell directly from any submenu  |
| <i>command [args..]</i> | When entering the name of a non-WXP shell command, the shell will execute the program <i>command</i> if that program's executable file is found in the search path defined by the "PATH" environment variable. The program will use any command line arguments <i>arg...</i> listed after the command. If the program is a WXP program, the selected menu will change to reflect the location of that program within the WXP submenus when the program exits. |

## Case Studies

The WXP shell has a case study interface which sets up resources (via environment variables) specific for each case study. You can list the case studies by running the "**case**" command:

```
*** Tornadoes ***
in_torn      - Indiana Tornados on (2 Jun 1990)
laf_torn     - Lafayette Tornado on (26 Apr 1994)
ill_torn     - Lafayette Tornado on (9 May 1995)
...
```

There is a header line which can be used to organize the case studies. The name listed (such as "in\_torn" is used with the **case** command to set the appropriate case. Once the case has been selected, the WXP prompt will change showing the case:

```
WXP-main> case 90_snow
Running Case: Cyclone Occlusion over Midwest (3 Dec 1990)

    Main Menu
    ...

WXP-main[90_snow]>
```

From this point forward, when you run WXP programs, they will point to data in the case study directory rather than the real-time data directory.

To go back to using real-time data, use the "**uncase**" command. The prompt will go back to normal reflecting the change.

**FILES**

**SEE ALSO**

# WXPCITY

## NAME

**wxpcity** - The WXP city database query program

## SYNOPSIS

**wxpcity** [*parameters...*] *value*

## PARAMETERS

| Command Line            | Resource                  | Default           | Description   |
|-------------------------|---------------------------|-------------------|---|
| <b>-h</b>               | <a href="#">help</a>      | No                | Lists basic help information.   |
| <b>-df=filename</b>     | <a href="#">default</a>   | <b>.wxpdef</b>    | Sets the name of the resource file.   |
| <b>-na=name</b>         | <a href="#">name</a>      | <b>wxpcity</b>    | Specifies the name used in resource file parsing.   |
| <b>-ba</b>              | <a href="#">batch</a>     | No                | Run program in batch mode   |
| <b>-me=level</b>        | <a href="#">message</a>   | <b>out2</b>       | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• text output - <b>print</b></li> <li>• type labels (ID=) - <b>out1</b></li> </ul>   |
| <b>-fp=filepath</b>     | <a href="#">file_path</a> | current directory | Specifies location of database files. All raw files used in this program are assumed to be in the <b>file_path</b> directory.   |
| <b>-in=input</b>        | <a href="#">input</a>     | <b>city</b>       | Specifies the input to the program: <ul style="list-style-type: none"> <li>• <b>city</b> - the <b>cities.raw</b> file</li> <li>• <b>cnty</b> - the <b>counties.raw</b> file</li> <li>• <b>zone</b> - the <b>zones.raw</b> file</li> <li>• <b>state</b> - the <b>states.raw</b> file</li> <li>• <b>dbase</b> - the <b>sao.cty</b> city database file</li> <li>• <b>input</b> - a specific raw file</li> </ul>  |
| <b>-ou=out[,out...]</b> | <a href="#">output</a>    | <b>all</b>        | Specifies what type of output is to be printed. This is based on the columns in the raw file. Here are some samples: <ul style="list-style-type: none"> <li>• <b>Name</b> - the long name of the city, county, etc</li> <li>• <b>Lat</b> - the latitude of the place</li> <li>• <b>Lon</b> - the longitude of the place</li> <li>• <b>ID</b> - the 4 letter ID of the station (cities only)</li> <li>• <b>WMO</b> - the 5 digit WMO number of the station (cities only)</li> <li>• <b>ST</b> - the 2 letter abbreviation of the state</li> <li>• <b>CN</b> - the 2 letter abbreviation of the country (dbase only)</li> <li>• <b>P</b> - the priority of the station (dbase only)</li> <li>• <b>Fip</b> - the FIPs county number of the place</li> <li>• <b>Zn</b> - the forecast zone number of the place</li> <li>• <b>ZWFO</b> - the WFO responsible for zone forecasts</li> <li>• <b>SWFO</b> - the WFO responsible for state forecasts</li> <li>• <b>WWFO</b> - the WFO responsible for warnings</li> <li>• <b>WFOs</b> - all possible WFOs (states only)</li> <li>• <b>Elev</b> - the elevation</li> <li>• <b>Pop</b> - the population (county only)</li> <li>• <b>Size</b> - the size of the place (county only)</li> <li>• <b>all</b> - all possible data for a place</li> </ul> <p>You can specify more than one type of output if types are</p> |

|                                   |                           |      |   |
|-----------------------------------|---------------------------|------|---|
|                                   |                           |      | separated by commas ( <b>Name, ID, ZWFO</b> )   |
| <b>-cf=cityfile</b>               | <a href="#">city file</a> | None | Specifies the name of the city database file used in finding station locations. If specified, parsing will be done with specified city database file rather than raw files. The <b>dbase</b> input option is automatically set.   |
| <b>-pa=param</b><br>[.param...]   | <a href="#">parameter</a> | None | Extra parameters: <ul style="list-style-type: none"> <li>• <b>first</b> - print only first occurrence</li> </ul>  |
| <i>value</i><br><i>type=value</i> | <a href="#">value</a>     | None | This is the particular place/station to search for. If none is specified, all places/stations are printed. If one is specified, it tests it against the <b>Name</b> , the <b>ID</b> and the <b>WMO</b> number. If the length of <i>value</i> is 4, it only checks the <b>ID</b> . For example, if you want data for Denver, you can enter <b>Denver</b> , <b>Denver_CO</b> , <b>KDEN</b> or <b>72565</b> (if you are using <b>dbase</b> ).<br><br>The value can specify what to parse against. Each of the types of information can be parsed for by preceding the requested value with the type. For example, to get all stations in Colorado, use <b>ST=CO</b> for value. |

## DESCRIPTION

This program can be used to get station information for either cities, counties or states based on plain English location names. The input to the program is either the name or the 2 or 4 letter ID for a particular city or state. The input name can be abbreviated as long as it uniquely defines the city, county or state desired. For example, to get information from Indianapolis or New York City:

```
wxpcity Indianapolis
Indianapolis 39.73 -86.27 KIND 72438 IN 097 047 KIND KIND KIND KIND 53 63 73 -5N
```

This will print all information about Indianapolis. If there is a problem with a city or county name appearing in more than one state, you can append the state ID:

```
wxpcity -in=cnty Marion_IN
Marion 39.78 -86.14 IN 097 047 KIND KIND KIND KIND 53 63 73 797159 1026
```

In this case, you cannot abbreviate the name. It must be the complete name for the city or county. To get specific information, limit the output with the **output** resource.

```
wxpcity -ou=zn Indianapolis
047
```

For stations with more than one word, use the underscore "\_" to separate all words.

```
wxpcity -ou=zwfo New_York
KNYC
```

To get more than one piece of information out of the database:

```
wxpcity -ou=Name, ID, ZWFO Indianapolis
Indianapolis KIND KIND
```

The individual fields are separated by spaces that can be used in shell scripts as arrays:

```
set stat=( `wxpcity Indianapolis` )
echo $stat[3]
```

You can select data based on specific fields. For example, you want all the stations from Indiana:

```
wxpcity -ou=Name,ID,ZWFO ST=IN
Evansville KEVV KIND
Fort_Wayne KFWA KIND
Indianapolis KIND KIND
Lafayette KLAF KIND
Muncie KMIE KIND
South_Bend KSBN KIND
Terre_Haute KHUF KIND
Bloomington KBMG KIND
Columbus KBAK KIND
Grissom KGUS KIND
Elkhart KEKM KIND
Gary KGYG KCHI
```

The value "**ST=IN**" says to search for type state (ST) and all stations where state matches "**IN**".

**NOTE:** Remember that the program will match any value/name/ID that matched the specified characters. If you don't specify a unique value/name/ID, more than one station will appear. As a result, it is often better to use the 4 letter ID to specify a station.

**NOTE:** Remember to replace spaces in a city or county name with underscores "\_". Otherwise, the query may not work properly.

This program was developed to help with writing enhanced shell scripts where city and state names are known. This is especially important for finding WSFO and WSO offices for parsing forecast and advisory information.

## FILES

- **cities.raw** - the cities information database file
- **counties.raw** - the county location database file
- **zones.raw** - the forecast zones database file
- **states.raw** - the states information database file
- **sao.cty** - the surface station location database file

## SEE ALSO

- [parse](#)
- [forecast](#)

# WXPDEF

## NAME

**wxpdef** - WXP resource query program

## SYNOPSIS

**wxpdef** [*parameters*] [*value*]

## PARAMETERS

| Command Line               | Resource                | Default             | Description  |
|----------------------------|-------------------------|---------------------|--|
| <b>-h</b>                  | <a href="#">help</a>    | No                  | Lists basic help information.  |
| <b>-df=<i>filename</i></b> | <a href="#">default</a> | <b>.wxddef</b>      | Sets the name of the resource file.  |
| <b>-na=<i>name</i></b>     | <a href="#">name</a>    | <b>wxddef</b>       | Specifies the name used in resource file parsing.  |
| <b>-ba</b>                 | <a href="#">batch</a>   | No                  | Run program in batch mode  |
| <b>-me=<i>level</i></b>    | <a href="#">message</a> | <b>out2</b>         | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• output values - <b>print</b> (default if resource specified on command line)</li> <li>• additional information - <b>out2</b></li> </ul> |
| <b>-ou=<i>output</i></b>   | <a href="#">output</a>  | none                | Specifies type of output. <ul style="list-style-type: none"> <li>• <b>full</b> - output full resource specification (as appears in resource file)</li> </ul>   |
| <i>resource</i>            | <a href="#">command</a> | none<br>User prompt | Specifies the resource to query.   |

## DESCRIPTION

The resource file is usually processed by the application programs to set initial program defaults, but in some cases, shell scripts may need to parse for these resource values. Also, users may wish to query resource specifications before running a particular application. Here are some examples:

To display all global defaults, use a ".", a period:

```
wxpdef .
*background: def
*con_path: /home/wxp/convert
*data_path: /home/wxp/data
*default: /home/wxp/etc
*file_path: /home/wxp/etc
*font_list: modern.fnt,gothic.fnt
*grid_path: /home/wxp/grid
*name_conv: ymdh
*raw_path: /home/wxp/raw
*watch_path: /home/data/ddplus
```

To display information for a particular resource, just list the desired resource. By default, only the resource value is displayed. If "full" output is desired, the resource will be displayed as it appears in the resource file. If more detailed information is needed, set the message level to "out2".

```
wxpdef data_path
/home/wxp/data

wxddef -ou=full data_path
*data_path: /home/wxp/data
```

```
wxpdef -me=out2 data_path
Name: *
Resource: data_path
Env. variable: wxpdata or wxpdata_path
Command line: -dp or -data_path
Conversion: path
Value: *data_path: /home/wxp/data (from /home/wxp/etc)
```

In the last case, the appropriate environment variable and command line parameters are listed. Finally, the value is listed with information about where the value was set, either resource file (filename listed) or environment variable.

To display information for a particular named resource, just list the name:

```
wxpdef hurricane.con_path
/home/web/hurricane
```

To show all resources for a particular program name, use ".", a period:

```
wxpdef xsat.
xsat.background: def
xsat.color_fill: 0-50
xsat.color_table: sat.clr
xsat.con_path: /home/wxp/convert
xsat.data_path: /home/wxp/data/mcidas
xsat.default: /home/wxp/etc
xsat.file_path: /home/wxp/etc
xsat.font_list: modern.fnt,gothic.fnt
xsat.grid_path: /home/wxp/grid
xsat.map_file: cont.dat,wxpstate.dat
xsat.name_conv: ymdh
xsat.raw_path: /home/wxp/raw
xsat.watch_path: /home/wxp/data
```

## FILES

- **.wxpdef** - The WXP resource file.

## SEE ALSO

# WXPFILE

## NAME

**wxpfile** - WXP file query program

## SYNOPSIS

**wxpfile** [*parameters*] [*value*]

## PARAMETERS

| Command Line        | Resource                 | Default            | Description   |
|---------------------|--------------------------|--------------------|---|
| <b>-h</b>           | <a href="#">help</a>     | No                 | Lists basic help information.   |
| <b>-df=filename</b> | <a href="#">default</a>  | <b>.wxpdef</b>     | Sets the name of the resource file.   |
| <b>-na=name</b>     | <a href="#">name</a>     | <b>wxpfile</b>     | Specifies the name used in resource file parsing.   |
| <b>-ba</b>          | <a href="#">batch</a>    | No                 | Run program in batch mode   |
| <b>-me=level</b>    | <a href="#">message</a>  | <b>out2</b>        | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>output values - <b>print</b> (default if resource specified on command line)</li> <li>additional information - <b>out2</b></li> </ul>  |
| <b>-cu=[hour]la</b> | <a href="#">current</a>  | None               | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.  |
| <b>-ho=hour</b>     | <a href="#">hour</a>     | None               | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.   |
| <b>-nh=num_hour</b> | <a href="#">num_hour</a> | <b>0</b> (no loop) | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.   |
| <b>-mo=model</b>    | <a href="#">model</a>    | None               | Specifies the model to use as input data. These are defined in the <b>model.lup</b> file. This file can define how model grids are pieced together. Some samples are: <ul style="list-style-type: none"> <li><b>ngm</b> -- Nested Grid Model</li> <li><b>eta</b> -- ETA model</li> <li><b>avn</b> -- Aviation model</li> <li><b>mrf</b> -- Medium Range Forecast model</li> </ul>   |
| <b>-ft=time</b>     | <a href="#">time</a>     | None               | This specifies the forecast time to use.  |
| <b>-le=level</b>    | <a href="#">level</a>    | None               | This specifies the vertical level to use. See the <b>level</b> resource for more information on what can be specified.  |
| <b>-va=variable</b> | <a href="#">variable</a> | None               | This specifies the variable to use.   |
| <b>-ou=output</b>   | <a href="#">output</a>   | none               | Specifies type of output. <ul style="list-style-type: none"> <li><b>file</b> - filename (default)</li> <li><b>hdrfile</b> -- ingest header filename</li> <li><b>date</b> -- date based on filename (<i>hhZ dd mmm yyyy</i>)</li> <li><b>valid</b> -- valid time label based on filename and forecast hour (<i>hhZ dd mmm yyyy</i>)</li> <li><b>label</b> -- label based on model, forecast time and filename</li> <li><b>year</b> -- the year based on filename (<i>yyyy</i>)</li> <li><b>mon</b> -- the month based on filename (<i>mm</i>)</li> </ul> |

|                     |                                 |  |  |
|---------------------|---------------------------------|--|--|
|                     |                                 |  | <ul style="list-style-type: none"> <li>• <b>month</b> -- the month based on filename (<i>mmm</i> as in <b>DEC</b>)</li> <li>• <b>day</b> -- the day based on filename (<i>dd</i>)</li> <li>• <b>hour</b> -- the hour based on filename (<i>hh</i>)</li> <li>• <b>seq</b> -- the sequence number (<i>yyyymmddhhMM</i>) based on filename.</li> <li>• <b>del</b> -- removes the file</li> <li>• <b>rm</b> -- removes the file</li> <li>• <b>exist</b> -- tests for file existence</li> <li>• <b>size</b> -- size of file</li> <li>• <b>next</b> -- the next file in the sequence</li> <li>• <b>prev</b> -- the previous file in the sequence</li> <li>• <b>label</b> -- a date/time label appropriate for use with plots. If the model and forecast time are specified, it will incorporate this into the label.</li> <li>• <b>time</b> -- the Unix time of the file (seconds since 1/1/1970)</li> <li>• <b>diff</b> -- the number of hours between file time and current.</li> <li>• <b>sdiff</b> -- the number of seconds between file time and current.</li> <li>• <b>mdiff</b> -- the number of minutes between file time and current.</li> <li>• <b>hdiff</b> -- the number of hours between file time and current.</li> <li>• <b>ddiff</b> -- the number of days between file time and current.</li> </ul> |
| <i>filename ...</i> | <a href="#"><u>filename</u></a> | None<br>User Prompt<br>Batch: current= <b>la</b> | The name of the input data file to be used.  |

## DESCRIPTION

This program displays and lists WXP file names.

## EXAMPLES

This prints file name plus path of the current converted surface file.

```
wxpfile -cu -if=sfc_cvt
/home/wxp/convert/17100197.cvt
```

This prints a date string for the current converted surface file.

```
wxpfile -cu -if=sfc_cvt -ou=date
17Z 1 OCT 1997
```

This determines whether the current converted surface file exists.

```
wxpfile -cu -if=sfc_cvt -ou=exist
exist
```

This prints the converted surface file name which corresponds to the sequence number 1991120617.

```
wxpfile -if=sfc_cvt #1997100117
/home/wxp/convert/17100197.cvt
```

This prints the number of hours between this file and current time.

```
wxpfile -if=sfc_cvt 12100197.cvt -ou=diff  
5
```

This prints the current raw upper air file.

```
wxpfile -cu -if=upa_dat  
/home/wxp/data/12100197.upa
```

This prints the next raw upper air file.

```
wxpfile -cu -if=upa_dat -ou=next  
/home/wxp/data/00100297.upa
```

With the use of the **num\_hour** resource, several files can be checked.

```
wxpfile -cu -nh=-5 -if=rad_cvt -ou=exist -me=out2  
/home/wxp/convert/13100197.sdc: exist  
/home/wxp/convert/14100197.sdc: not found  
/home/wxp/convert/15100197.sdc: exist  
/home/wxp/convert/16100197.sdc: exist  
/home/wxp/convert/17100197.sdc: exist
```

This prints the current NGM GRIB file.

```
wxpfile -cu -if=grib_ngm -na=grbcalc  
/home/wxp/hds/12100197.gbn
```

This prints a date/time label for the current surface converted file.

```
wxpfile -cu -if=sfc_cvt -ou=label  
Analysis for 17Z 1 OCT 97
```

This prints a date/time label for the MRF 7 day grid file.

```
wxpfile -cu -if=grib_eta -mo=eta -ft=h24 -ou=label  
24 hour ETA valid 12Z FRI 3 OCT 97
```

## FILES

## SEE ALSO

# WXPLOOP

## NAME

**wxplloop** - WXP loop and overlay program

## SYNOPSIS

**wxplloop** [*parameters*] [*files...*]

## PARAMETERS

| Command Line                             | Resource                    | Default                          | Description   |
|--|-----------------------------|----------------------------------|---|
| <b>-h</b>                                | <a href="#">help</a>        | No                               | Lists basic help information.   |
| <b>-df=filename</b>                      | <a href="#">default</a>     | .wxpdef                          | Sets the name of the resource file.   |
| <b>-na=name</b>                          | <a href="#">name</a>        | wxplloop                         | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                               | <a href="#">batch</a>       | No                               | Run program in batch mode   |
| <b>-me=level</b>                         | <a href="#">message</a>     | out2                             | Specifies level of messages to be displayed   |
| <b>-wn=window</b>                        | <a href="#">window_num</a>  | 0                                | Specifies the window number. This sets up a unique route between this program and WXPloop so that WXPloop can be run multiple times simultaneously. |
| <b>-pd=domain</b>                        | <a href="#">plot_domain</a> | None                             | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.   |
| <b>-cd=commands</b>                      | <a href="#">command</a>     | None                             | Specifies a list of commands to execute.  |
| <b>-pp=param</b><br>[ <i>,param...</i> ] | <a href="#">plot_param</a>  | None                             | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.   |
| <b>-fl=font_list</b>                     | <a href="#">font_list</a>   | modern.fnt                       | Specifies a list of fonts to use in the plots.  |
| <b>-ct=color_table</b>                   | <a href="#">color_table</a> | wxp.clr                          | Specifies the color table to use in the plot. This maps color names to actual RGB color values.   |
| <b>-ti=title</b>                         | <a href="#">title</a>       | Program name                     | Specifies the window title.   |
| <b>-ic</b>                               | <a href="#">icon</a>        | Off                              | Specifies whether to open window in iconified mode.   |
| <b>-bg=color</b>                         | <a href="#">background</a>  | black<br>white for inverse plots | Specifies the background color of the window.   |
| <b>-ge=geometry</b>                      | <a href="#">geometry</a>    | 640x480                          | Specifies the window and paper geometry.  |
| <i>files ...</i>                         | <a href="#">filename</a>    | None                             | Specifies the files to load into the WXPloop window. Specifying more than one will create a loop.   |

## DESCRIPTION

The WXPloop program is a program set up to allow multiple WXP programs to overlay graphics and produce loops. The program offers a window to other WXP programs so that their graphics can be drawn to that window. Also the program uses the concept of a **pixmap** which is an allocated portion of memory that emulates a window to create pseudo windows which can be looped or animated. Once WXPloop runs, a single main window can be opened, but several pixmaps can be also be created (up to 128). Sequencing through these pixmaps will give the sense of animation.

Information can be drawn onto the window or a pixmap in one of three ways:

- using the annotation features of WXPloop to draw lines, text and markers
- using the pixmap ID to allow another program to draw directly to that window or pixmap
- importing an image file into the loop

WXPloop can be run in one of two operational modes: interactive and batch

## Interactive Mode

Interactive mode allows the user to communicate with the WXPloop program, issuing commands to it via the [loopset](#) program. Once WXPloop is started, a message queue is opened allowing communications with the **loopset** program. This message queue key ID is a positive integer number set by the [window\\_num](#) resource. More than one WXPloop program can be running but each must have a unique **window\_num**. If a queue exists for a particular window number and WXPloop is run, the program will exit with a message "Unable to create message queue". At this point, either the queue must be deleted or another window number must be picked. The "**loopset alloc**" command will return the next available window number after 20.

The program is initially in STOP mode and runs any commands passed through the command line ([command](#) resource). Then, the program will load into a loop, any image files specified on the command line. After the images are loaded, the program animates those images and waits for commands to be entered via **loopset** where pixmaps can be created and deleted, images annotated and loop parameters changed.

At any point through **loopset**, a window ID of the main window can be queried. This window ID can then be passed to any WXP graphics program by specifying the window device.

Example return: **+8627531:8627533:800x600** or **+8627531:8627533** (if size is the default 640x512)

Also, pixmaps can be created. The pixmap ID can be queried and graphics drawn directly to the pixmap by specifying its ID and size to the device resource.

## Batch Mode

WXPloop can also be run in a batch mode. If the following command is entered:

```
wxploop -wn=-1 -ba
```

By specifying a window number of **-1**, the WXPloop program does not open up a message queue. This alleviates the problem of having to allocate a different message queue for each WXPloop window. Obviously, this lack of interactivity limits the capability of WXPloop. This configuration is optimal for simple overlay displays and cannot be used for animation since there is no way to create pixmaps with a shell script.

The program will run those commands pass on the command line and load any images in listed on the command line. The WXPloop program opens the main window and titles the window "**wxp\_win**". The program will then print the window ID of the main window allowing WXP programs to access that window with the window specification under the **device** resource.

**+8627531:8627533:800x600**

Then the WXPloop program will automatically background itself. This way, the window ID can be set to a shell variable and passed to another graphics program. Manual backgrounding is not necessary with "&" on the command line and is also not recommended.

The window ID information can also be written to a file if the filename is specified with the **window\_num** resource "**-wn=filename**". The filename is then specified with the device resource "**-de=w,,filename**".

## Keyboard and Mouse Controls

### STOP GRAPHICS MODE

Once WXPloop is started, it defaults to STOP mode. In this mode, just the main X window is active. No animation is possible. Different images in the loop can be viewed, however. Some important mouse and keyboard commands are:

| Keyboard/Mouse | Command                    |
|----------------|----------------------------|
| <RightButton>  | Display next image in loop |

|                |                                |
|----------------|--------------------------------|
| <LeftButton>   | Display previous image in loop |
| <MiddleButton> | Toggle to LOOP mode            |
| <Meta>d        | Delete current image in loop   |
| <Meta>z        | Zoom image around cursor 2x    |
| <Meta>g        | Saves window to a GIF file     |
| <Meta>x        | Saves window to a XWD file     |
| <Meta>q        | Quit WXPloop                   |
| <Enter>        | Quit WXPloop                   |

The cursor is a circle (X11) or arrow (Win32).

### LOOP GRAPHICS MODE

In this mode, pixmaps are copied into the main window in a set order. Any information previously displayed to the main window will be destroyed. Some important mouse/keyboard commands are:

| Keyboard/Mouse | Command                                    |
|----------------|--|
| <RightButton>  | Reduce delay between pixmaps by .1 seconds |
| <LeftButton>   | Increase delay by .1 seconds               |
| <MiddleButton> | Toggle to STOP mode                        |
| <Meta>z        | Zoom image around cursor 2x                |
| <Meta>g        | Saves all pixmaps to GIF files             |
| <Meta>x        | Saves all pixmaps to XWD files             |
| <Meta>q        | Quit WXPloop                               |
| <Enter>        | Quit WXPloop                               |

The cursor is two arrows pointing beginning to end (X11 exchange cursor) or an up/down arrow (Win32). Zooms in loop mode are done on the fly to preserve the integrity of the original image.

### Loopset Program

The loopset program allows WXPloop's parameters to be changed. A IPC channel is used to communicate with the ingest program. When the **loopset** program establishes a connection with **wxploop**, the following prompt will appear:

```
Message queue opened (id:nnnnn)
Connected to WXPloop program
Type help for help
Message:1:
```

At this point, you may enter a WXPloop command. To see the list of WXPloop commands, check the [wxploop](#) command documentation. The **loopset** program continues to accept dataset commands until "**quit**" or "**exit**" commands are entered. If a valid *command* is specified from the resource file or command line, only that command will be processed and the program will exit.

### Drawing

WXPloop allows the user to annotate images using the drawing commands available through loopset. These allow the user to draw lines, plot markers or text. There is a full range of set attribute commands to change colors, styles, sizes, etc. The drawing canvas is based on the viewport and window concept. There are 3 defined transformations that can be used:

- 0 - Pixel coordinate system (0-width,0-height) where 0,0 is in the upper left corner.
- 1 - Normalized coordinates (0-1, 0-1) where 0,0 is at the lower left and the extent is 0 to 1 for the largest side and a value less than 1 for the smaller side.
- 2 - Map coordinates where the mapping is based on the **domain** specified either through the **plot\_domain** resource or through the "**set domain**" command.

## Zooming Images

Images can be zoomed in either STOP or LOOP mode with **<Meta>z**. Each zoom enlarges the image at the cursor position 2x. There is a limit of 1024x on enlargement. In loop mode, the zoom is performed before each image is displayed in the window. These are not cached because contents of original unzoomed window may be changing during the zoom. Also, this avoids the extra memory requirements necessary to cache all the zoomed images.

## Colors

WXPloop starts with no defined colors. If a color table is specified with the **color\_table** resource, it is loaded into the program and those colors fixed. To use any other colors, a color table must be set up by directing WXPloop to read in a color table file with the **"color"** command. This will first, allocate those colors so that they cannot change if another program is run and second, set up those colors to be used by the internal WXPloop annotation commands.

**NOTE:** When plotting graphics to a WXPloop window or pixmap, it is highly recommended to allocate all colors used in the final graphic by using the **"color"** command. If this is not done, no colors are allocated and therefore fixed in the server's color table. Even though each WXP program allocates its own colors, these only remain allocated for the duration of each graphics program which is independent of WXPloop. Once the graphics program exits, the colors are deallocated and those colors displayed on a WXPloop window or pixmap may be changed by the next program that wishes to allocate colors. Running the **"color"** command in WXPloop will allocate/lock those colors in the color table. If a graphic uses multiple color table, the **"color"** must be run again to allocate the other color tables. This command does not deallocate any previous colors.

## Saving Loops

Once a loop has been created, it can be saved to a GIF (Graphics Interchange Format used by Compuserve -- Default) or X window dump files with the **save** command. The specified filename is sent through the file name convention function. Keys include **"%p"** for program name and **"%i"** for frame index. Then these files can be read back into WXPloop later on and again animated. WXPloop will read and write image files for 1,4, 8, 16 and 24 bit displays. WXPloop will read any image files as long as the display WXPloop is connected to has a depth greater than or equal to the depth of the image contained in the file.

## COMMANDS

- **clear image**  
Clears the image in pixmap *image*. This blanks the pixmap returning it to an all black image. Image can be **"win"** for the window, **"last"** for the last image of a loop or a image number **"#"**. If no image is specified, the current image is cleared.
- **clear**  
Clears the X window. This only clears the window and not any of the pixmap images.
- **colortable filename**  
Loads the color table in from the file *filename* and allocates the colors on the X server default colormap. This is a standard WXP color table file.

**NOTE:** It is a very good idea to load the appropriate colormap prior to running any WXP application in order lock the colors into the server. Failing to do so could result in losing color definition.

- **create**  
Creates a new pixmap to which data may be drawn. This pixmap is appended to the end of the current loop. If there are no images currently in the loop, the program will go from STOP mode to LOOP mode.
- **create pixmap**  
Creates a new pixmap just as the **create** command does. The difference is that the response is the window ID, pixmap ID and size.

Example: **+8627531:8627533:800x600** or **+8627531:8627533** (if size is the default 640x512).

- **delete image**

Deletes the current image and pixmap from the loop. All remaining images are moved up in the loop one position. Image can be "**last**" for the last image of a loop or a image number "#". If no image is specified, the current image is deleted.

- **goto image**  
Goes to image number *image*. Image can be "**win**" for the window, "**last**" for the last image of a loop or a image number "#".
- **kill**  
This kills the WXPloop program. This involves freeing all the pixmaps and graphics contexts, destroying the window, deleting the message queue and exiting the WXPloop program.
- **line x1,y1 x2,y2**  
Draws a line from x1,y1 to x2,y2.
- **load image filename [x,y]**  
Loads an image from *filename* into image *image*. If an image already existed at that location, it is overwritten. An optional title may be defined. These titles are useful in labeling the image as to its contents. The colors are compressed and allocated (fixed) before copying to *image*. Image can be "**win**" for the window, "**last**" for the last image of a loop or a image number "#". An optional location can be specified.
- **loop**  
Toggles the loop from looping to stopped.
- **mark x,y**  
Draws a marker at x,y
- **open [name]**  
Opens the window frame for drawing and plotting. The window **must** be opened prior to any plotting or errors occur. The window name may be specified with *name*.
- **overlay image filename [x,y] [back]**  
Overlays an image from the file *filename* on top of image *image*. A bitmap is created based on all BLACK pixels contained in the overlay image, so that all areas in the overlay image that were black are now transparent. The colors in the overlay image are compressed and allocated (fixed) before overlay. Image can be "**win**" for the window, "**last**" for the last image of a loop or a image number "#". An optional location can be specified. Also, a background color can be specified to set transparency.
- **query pid**  
Returns the process ID of the WXPloop process.
- **query last**  
Returns the index of the last image in the loop
- **query num**  
Returns the number of images in the loop.
- **query window\_id**  
Returns the base window ID, draw window ID and size.

Example: +**8627531: 8627533:800x600** or +**8627531: 8627533** (if size is the default 640x512).

- **quit**  
Exits loopset program.
- **save image filename**  
Saves the image *image* to the file *filename* in the specified image file format (XWD or GIF). Image can be "**win**" for the window, "**last**" for the last image of a loop, "**all**" for all images in a loop or a specific image number "#". The filename is sent through the file name convention function. Keys include "%p" for program name and "%i" for frame index.
- **set delay image delay**  
Sets the loop delay in seconds. This will change the image end delay for all images in the loop except for the last image whose delay is set by **set pause**. Image can be "**win**" for the window, "**last**" for the last image of a loop or a image number "#". If no image is specified, the current image is set. (Default: 0.5 seconds)
- **set direction backward**  
Sets the loop direction to backward or essentially reverses the flow of the loop. In this case, the loop will proceed through the loop from highest image number to lowest.
- **set direction forward**

Sets the loop direction to forward. In this case, the loop will proceed through the loop from lowest image number to highest.

- **set direction alternating**  
Sets the loop direction to alternating. In this case, the loop will proceed through the loop from lowest image number to highest and then reverse going highest to lowest and so on.
- **set domain *domain***  
Sets the plotting domain for the loop. This should correspond to the value of the **plot\_domain** resource. This may be queried at a later time for applications wishing to overlay information or add new images to the loop.
- **set draw *image***  
Sets the drawing window or pixmap for all annotation. If the draw window is the main window, *image* is "**window**" (DEFAULT), otherwise, it is the desired image number. This is useful for scripts that need to annotate on multiple pixmaps. Image can be "**win**" for the window, "**last**" for the last image of a loop or a image number "#". If no image is specified, the current image is set.
- **set line color *color***  
Sets the line color to a color name. All subsequent lines drawn by the user will be in this color.
- **set line style *style***  
Sets the line style to *style*. Valid styles are **solid,dash,doubledash**.
- **set line width *width***  
Sets the line width to *width* pixels.
- **set mark color *color***  
Sets the marker color to an established color name. All subsequent lines drawn by the user will be in this color.
- **set mark size *size***  
Sets the marker size to *size*.
- **set mark style *style***  
Sets the marker style to *style*. Valid styles are **pnt,plus,star,circle,x,square,tri,itri,dot**
- **set mark width *width***  
Sets the marker width to *width* pixels.
- **set name *name***  
Sets or changes the window name which is displayed in the title bar of the X window.
- **set output *type***  
Sets the output file type to **gif** (Default) or **xwd**.
- **set pause *pause***  
Sets the delay in seconds for the last image in the loop. (Default: 1.0 seconds)
- **set size *widthxheight***  
Sets the size of the window to be used by WXPloop. (DEF: **640x512**)
- **set text alignment *align***  
Sets the text alignment. The alignment is a 2 character sequence that specifies horizontal and vertical alignment. The first character is horizontal which can be either **left**, **center** or **right**. The second character is the vertical alignment which can be **upper**, **top**, **center**, **base** or **lower**. All subsequent text entered by the user will be drawn left, center, or right aligned to the starting position.
- **set text color *color***  
Sets the text color to an established color name. All subsequent text entered by the user will be in this color.
- **set text expan *expansion***  
Sets the text expansion factor to *expansion*. This specifies the relative horizontal size of the text.
- **set text font *name***  
Sets the text font to *name*. The font name is either the alias in the **wxp.fnl** file or the font number, The font is read in and used as the font for all subsequent text commands. The default font is "**modern.fnt**".
- **set text height *height***  
Sets the text height to *height* pixels. This specifies the actual height of the text from baseline to cap.
- **set text width *width***  
Sets the text width to *width* pixels. This specifies the width of the lines used to draw the text. Setting this to a value greater than 1 makes the text bold.
- **set viewport *x1,y1 x2,y2***

Sets the viewport extent to  $x1,y1$   $x2,y2$ . This is the limits of the plotting area and is expressed as a number between 0 and 1. This is also referred to as the clipping rectangle.

- **set window**  $x1,y1$   $x2,y2$   
Sets the window coordinates to  $x1,y1$   $x2,y2$ . This is the actual plotting coordinate system that will be mapped to the viewport.
- **start**  
Starts the loop.
- **step**  
Single steps the loop.
- **stop**  
Stops the loop.
- **text**  $x,y$  *text*  
Displays the text *text* on the window at the position  $x,y$ . The text is centered at this position.
- **update**  
Updates the window. Use this after generating a graphic to the window.

## FILES

## SEE ALSO

- [loopset](#) -- WXP loop parameter setting program

# XSAT

## NAME

**xsat** - Satellite image display program

## SYNOPSIS

**xsat** [*parameters...*] *filename*

## PARAMETERS

| Command Line                 | Resource                   | Default           | Description   |
|------------------------------|----------------------------|-------------------|---|
| <b>-h</b>                    | <a href="#">help</a>       | No                | Lists basic help information.   |
| <b>-df=filename</b>          | <a href="#">default</a>    | .wxddef           | Sets the name of the resource file.   |
| <b>-na=name</b>              | <a href="#">name</a>       | xsat              | Specifies the name used in resource file parsing.   |
| <b>-ba</b>                   | <a href="#">batch</a>      | No                | Run program in batch mode   |
| <b>-me=level</b>             | <a href="#">message</a>    | out2              | Specifies level of messages to be displayed <ul style="list-style-type: none"> <li>• file information - <b>mess</b></li> <li>• base image information - <b>out2</b></li> <li>• extra image information - <b>out3</b></li> </ul>   |
| <b>-fp=filepath</b>          | <a href="#">file_path</a>  | current directory | Specifies location of database files.   |
| <b>-dp=datapath</b>          | <a href="#">data_path</a>  | current directory | Specifies the location (path) of the input raw data files. This may be modified in the name convention file.  |
| <b>-cp=conpath</b>           | <a href="#">con_path</a>   | current directory | Specifies the location (path) of the input converted data files. This may be modified in the name convention file.  |
| <b>-ip=imagepath</b>         | <a href="#">image_path</a> | current directory | Specifies the default location of the output GIF images generated by the program.   |
| <b>-nc=name_conv</b>         | <a href="#">name_conv</a>  | name_conv         | The name convention file specifies how files are named in WXP. This sets which name convention file to use.   |
| <b>-inp=input</b>            | <a href="#">input</a>      | area              | Specifies the input file type. The default is <b>area</b> . The types are: <ul style="list-style-type: none"> <li>• <b>area</b> - AREA file</li> <li>• <b>img</b> - WXP image file. Name convention set to <b>sat_img</b>.</li> <li>• <b>awip</b> - NOAAPORT image file</li> <li>• <b>unisat</b> - Unisys image file</li> </ul>   |
| <b>-if=in_file</b>           | <a href="#">in_file</a>    | sat_vis           | Specifies the input file name tag. The default is <b>sat_vis</b> , but it can be modified to any value in the filename convention file. A full name convention can be specified as well. Some special types are: <ul style="list-style-type: none"> <li>• <b>vis</b> - Name convention is <b>sat_vis</b></li> <li>• <b>ir</b> - Name convention is <b>sat_ir</b></li> <li>• <b>wv</b> - Name convention is <b>sat_wv</b></li> <li>• <b>img</b> - WXP image file, name convention is <b>sat_img</b></li> <li>• <b>awip</b> - NOAAPORT image file, name convention is <b>sat_vis</b></li> <li>• <b>unisat</b> - Unisys image file.</li> </ul> |
| <b>-of=out_file[,out...]</b> | <a href="#">out_file</a>   | None              | Specifies what type of image file to create. By default, no output file is generated when the program is run. To produce an output file, this resource must be specified. There are the following output file types: <ul style="list-style-type: none"> <li>• <b>img</b> - WXP image file</li> </ul>  |

|   |                             |   |  |
|---|-----------------------------|---|--|
|   |                             |   | <p>Second, the <b>out_file</b> resource specifies the output file name tag. The default is <b>sat_img</b> but it can be modified to any value in the filename convention file. The file name tag can be specified separately or appended to the file type:</p> <pre>img:sat_img_ir</pre> <p>A full filename can be specified which can contain name convention tags.</p> |
| <b>-pf</b> = <i>file_param</i>                    | <a href="#">file_param</a>  | User prompt<br>Batch: <b>use</b>  | <p>This specifies how to treat output files that exist:</p> <ul style="list-style-type: none"> <li>• <b>Use</b> - use the existing file, don't overwrite. The program terminates when this occurs.</li> <li>• <b>Over</b> - overwrite the existing file</li> <li>• <b>Y</b> - same as overwrite</li> <li>• <b>N</b> - same as use.</li> </ul>                            |
| <b>-mf</b> = <i>mapfile</i> [,...]                | <a href="#">map_file</a>    | <b>wxp.map</b> or <b>cont.dat</b> ,<br><b>country.dat</b> for domains outside North America | Specifies the name of the map database file. Multiple maps can be specified along with attributes to describe how to draw the map.   |
| <b>-cu</b> =[ <i>hour</i>   <b>la</b> ]           | <a href="#">current</a>     | None  | This specifies to use current data files. The current filename is based on the name convention. An optional <i>hour</i> can be specified for older data. If <b>la</b> is specified, the program will search back to find the most recent available file.   |
| <b>-ho</b> = <i>hour</i>                          | <a href="#">hour</a>        | None  | This resource specifies the exact hour that a data file is valid for. This locks in the start hour for a multi-file sequence.  |
| <b>-nh</b> = <i>num_hour</i>                      | <a href="#">num_hour</a>    | <b>0</b> (no loop)  | This specifies the number of hours that will be used in the plot. If this is not specified, a single hourly plot will appear. If this is specified, a loop of plots over this time will be generated.  |
| <b>-pd</b> = <i>domain</i>                        | <a href="#">plot_domain</a> | User prompt<br>Batch: first domain in <b>wxp.reg</b> file.                                  | This specifies the plotting domain. The projection, aspect ratio and the domain size are specified.  |
| <b>-re</b> = <i>region</i>                        | <a href="#">region</a>      | Same as plot domain   | This resource is synonymous with the <b>plot_domain</b> .  |
| <b>-gd</b> = <i>grid_domain</i>                   | <a href="#">grid_domain</a> | None  | This specifies the domain to use for an un navigated image. This only affects images without navigation. If this is not specified, the navigation is based on pixel values.  |
| <b>-sc</b> = <i>scale_factor</i>                  | <a href="#">plot_scale</a>  | <b>1.0</b>  | Specifies a scaling factor for all plotted text, symbols, vectors, etc.  |
| <b>-fm</b> = <i>format</i>                        | <a href="#">plot_format</a> | None  | Specifies the output format of the plots. This is generally used to specify how many decimal places get displayed on the plot and in the output raw files.   |
| <b>-pp</b> = <i>param</i><br>[, <i>param</i> ...] | <a href="#">plot_param</a>  | None  | Specifies additional plotting parameters. See the <a href="#">plot_param</a> resource for more details.  |
| <b>-dr</b> = <i>drawable</i> [,...]               | <a href="#">draw</a>        | <b>all</b>  | Specifies which part of the plot is to be drawn.   |
| <b>-la</b> = <i>label</i> [,...]                  | <a href="#">label</a>       | Program defaults  | Specifies the labels to be plotted around the plot. By default, labels are generated by the program and plotted above and below the plot. This resource overrides these labels.  |
| <b>-fl</b> = <i>font_list</i>                     | <a href="#">font_list</a>   | <b>modern.fnt</b>   | Specifies a list of fonts to use in the plots.   |
| <b>-ct</b> = <i>color_table</i>                   | <a href="#">color_table</a> | <b>sat.clr</b>  | Specifies the color table to use in the plot. This maps color names to actual RGB color values.  |

|  |                            |  |   |
|--|----------------------------|--|---|
| <b>-cot=</b> <i>color[:attr]</i>           | <a href="#">color text</a> | <b>white</b><br><b>black</b> for inverse plots.                | Specifies the text color for labels above and below the plot.   |
| <b>-com=</b> <i>color[:attr]</i>           | <a href="#">color map</a>  | <b>red</b>   | Specifies the color for background maps.  |
| <b>-coln=</b> <i>color[:attr]</i>          | <a href="#">color line</a> | <b>magenta:st=dsh</b>  | Specifies the color for lat/lon lines.  |
| <b>-cof=</b> <i>[val:]color[:attr],...</i> | <a href="#">color fill</a> | A predefined color set based on the <b>wxp.clr</b> color table | Specifies the pixel to color set. This is also used for satellite image enhancement. See the <b>color_fill</b> resource for more details. |
| <b>-de=</b> <i>device[,...]</i>            | <a href="#">device</a>     | User prompt<br>Batch: <b>display</b>                           | Specifies the output device.  |
| <b>-ti=</b> <i>title</i>                   | <a href="#">title</a>      | Program name   | Specifies the window title.   |
| <b>-ic</b>                                 | <a href="#">icon</a>       | <b>Off</b>   | Specifies whether to open window in iconified mode.   |
| <b>-bg=</b> <i>color</i>                   | <a href="#">background</a> | <b>black</b><br><b>white</b> for inverse plots                 | Specifies the background color of the window.   |
| <b>-ge=</b> <i>geometry</i>                | <a href="#">geometry</a>   | <b>640x480</b>   | Specifies the window and paper geometry.  |
| <i>filename[#seq]</i>                      | <a href="#">filename</a>   | None<br>User Prompt<br>Batch: <b>current=la</b>                | The name of the satellite image file to be plotted. An optional sequence number can be added to designate the time for non-WXP files.     |

## DESCRIPTION

This program plots satellite imagery and overlays a map. The input to the program is either a McIDAS AREA file, NOAAPORT image or a WXP image file.

### McIDAS Area files

Area files come in various sizes and resolutions. The image file includes a navigation block to aid programs in determining the location of the image so that data may be overlaid on the image. This navigation block contains the orbit information of the satellite which is used to locate the satellite over the earth's surface and the information needed to translate scan lines and image elements to locations on the earth's surface. Since the image is navigated, it can be remapped. The image retains the original digital information so that enhancements may be applied and temperatures may be inferred from the infrared images.

### NOAAPORT image files

NOAAPORT images are preset image files that are formatted similar to GRIB files. There is a product definition block which defines the satellite, image type, and projection. The image then follows as a sequence of 8 bit pixel values which can be enhanced using any of the standard enhancement techniques. The image can be navigated and remapped. To use NOAAPORT images, specify "**awip**" for the **input** resource.

### Unisys image files

This is a generic image format used by Unisys for transmission of satellite imagery. In most cases, these are sectorized images with no navigation. Pseudo-navigation can be added with the [grid domain](#) resource. Some of the images are remapped to a Lambert Conformal projection and these can be remapped. The image contains a small header in ASCII which describes the product. The 8 bit image data follows.

### WXP image files

These are general purpose image files. This contains 8 bit image data plus a navigation block equivalent to the projection grid given in grid files. Since the image is navigated, it can be remapped. The image has 8 bits of resolution and may be enhanced with any of the standard enhancement techniques. To use WXP images, specify "**img**" for the **input** resource.

Then, based on an input image type and a plot domain, a plot is generated. Overlay plots and loops can also be generated if the appropriate options are set.

The programs starts off by prompting the user for input satellite file name. The user may specify the input file either via the command line or through the [current](#) resource. The input naming convention is specified by the `sat_vis` tag but this can be changed with the [in\\_file](#) resource. To generate a loop, the [num\\_hour](#) resource can be specified. This will tell the program to loop through this number of hours.

## Image Navigation

Most satellite images are navigated. This means they contain information that will allow a program to map pixel locations to latitude, longitude coordinates. Once this is known, a program can overlay a map, zoom to a specific location or totally remap the image to a new projection. In some cases, images are un-navigated. In this case, the [grid\\_domain](#) resource can be used to specify a pseudo-domain which can be used to overlay maps or remap the image. It is up to the user to determine this domain and is done by trial and error. Once determined and if the image does not change, this domain can be added to the `wxp.reg` file for future use.

## Remapping

By default, images are not remapped but instead clipped to fit the window size. There is no attempt to resize the image. This means most images cannot be fully displayed in the window. There are two methods for displaying more of the image using the [plot\\_domain](#) resource:

- **zoom:clat:clon:factor**  
This will zoom the image by the factor listed centered on the specified `clat,clon`.
- **full plot domain with map projection**  
By specifying a full plot domain, the image will be remapped to that domain. Region keywords can also be used.

If the [plot\\_domain](#) is a map projection, then the image will be remapped. This is done by resampling the image using a bi-linear interpolation scheme to compute new pixel values. The location of the pixel can be computed in a number of ways:

- **tran\_sat** - this uses the internal navigation information to compute a lat,lon location for each pixel and remap it to a exact location in the original image. This can be time consuming since the internal navigation uses complex algorithms to compute satellite orbit parameters.
- **tran\_wxp** - this uses the internal navigation information only to compute a WXP plot domain which is then used to remap the image. This is faster for remapping since the algorithms are simpler, but this may not produce an exact remap. The edges of the image may have a map that doesn't match up with the image. For smaller domains, this works well.
- **remap[=len]** - this uses the more accurate transformations only every 10 pixels. A linear interpolation is used in between transformed pixels. This speed up the remap process considerably (by a factor of 4-8). The default is every 10 pixels which works for most projections but this length can be changed if needed. Using a length greater than 10 can improve speed but reduce accuracy. This is only recommended for some satellite projection remappings.

Once completed, all navigation information is based on the new image.

## Zooming

After remapping is complete, the image may be zoomed (enlarged, reduced). A zoom factor may be specified with the `zoom=factor` plot parameter or with the `zoom` domain projection. The zoom domain syntax is `zoom,clat,clon,factor` where the `clat` and `clon` are used to center the zoomed image on a particular spot. The zoom factor uses values greater than one to specify an enlargement and values less than one to specify a reduction. In other words, to get a 1200x600 image into a 640x512 window, a zoom factor of .6 might be used. Enlargements are done by using bilinear pixel averaging.

## Color Table

Once the image is remapped and zoomed, a color table is applied to it. This is based on the [color\\_table](#) and the [color\\_fill](#) resources. The recommended setup is to use the `sat.clr` color table with `0-50` as the color fill specification.

This results in a image using 50 shades of gray. Other table/fill combinations can be used to modify/enhance the image. These are the lines for the resource file:

```
xsat.color_table: sat.clr
xsat.color_fill: 0-50
```

The original pixel values are modified from a 8 bit intensity to an even range of colors in the is modifies the original pixel values to color indices for plotting.

## Dithering

The image may also be dithered for display on displays which have depth less than 8 bits. The image may also be set up for overlay on top of another image where black regions on the image become transparent allowing the underlying image to show through. The colors can be specified with dither value and color. This is handy for screens that cannot support gray shades such as monochrome displays. Dither colors are specified by using the following syntax:

**black%6white**

The original or background color is specified followed by a "%", then the dither percent value and the dither or foreground color. The above example uses the 3x3 dither matrix and gives possible dither values of 0 through 9. The result is a 6/9 dither of white over black. Using 0 for the dither value gives the original color. Using 9 for the value gives the only dither color. This is somewhat restrictive for monochrome screens, so a second dither scheme is provided which is a 5x5 dither matrix. This gives a total of 26 possible dither values which are specified by lower case alphabetic characters "a" through "z". The value "a" gives the original color and "z" gives the dither color.

## Enhancement

If needed, the image may be enhanced. Satellite enhancement is conceptually similar to the color fill parameters used by other WXP programs except attributes are not applicable for satellite pictures. The color parameters are specified with the **color\_fill** resource. The syntax is:

```
[val:]color[%dither],[val:]color[%dither]..
```

The color values *val* are image pixel values ranging from 0 to 255. The value may be specified as a image pixel value or as a infrared temperature based on the GOES conversion scheme. The temperature is specified by preceding the value by a "c". For example, "c-25" specifies the image pixel value corresponding to -25 Celsius. This way infrared images can be enhanced based on the temperatures the image represents.

**NOTE:** This works only for increasing image pixel values. As the image pixel values increase, the temperatures DECREASE and therefore temperature values must be listed in decreasing order.

## Enhancement Files

Since enhancement schemes can become very complicated, specifying them on the command line would be inappropriate. Thus, an enhancement file should be used. This file is very similar to a color fill file except it has an additional line at the beginning of the file that names the enhancement. Here is the MB enhancement file:

```
MB
0:black
c29:1
2-9
c6:10
11-15
c-31:16
c-32:cyan
c-43:lcyan
c-54:lblue
```

```

c-60:blue
c-64:green
c-70:lgreen
c-76:brown
c-81:yellow
c-90:black

```

## Overlay Images

If satellite overlay is specified (**over** plot parameter), the image will be placed on the screen as a non-destructive overlay for all parts of the image where the pixels are colored **black**. These areas will then be transparent whereas the non-black areas will be opaque. The **color\_fill** resource is then used to specify which pixel values are transparent. Also, a fill pattern can be applied to the overlay image using the **fill[=*pattern*]** plot parameter resource.

## Final Plot

The image is then converted into image which can be displayed within the window. The final image size is default 640x427 pixels depending on domain and window geometry. If the image is navigated, a map is then overlaid to provide proper georeferencing. The projection information is then displayed so that it can be used with overlay plots. In most cases, it is recommended to remap the image since many WXP programs don't work well overlaying to a satellite projection. The program finally annotates the image with the date and time in the upper right and general comments about image type in the upper left. For area files, the comment line in the area header is displayed by default. This can be rather cryptic so it is recommended to change the label with the **label** resource (-**la=ul:GOES\_East\_IR**).

The image may also be printed using a Postscript interface built into the **xsat** program. This creates a gray scale image based on a 640x512 sized window. If color output is requested, specify **pscolor**. This requires a Postscript level 2 printer (for the **colorimage** Postscript command) which all color Postscript printers support. Navigation is used to overlay a map for final output. Printer output is not prompted for but must be specified on the command line. It should also be noted that images can take over ten minutes to print on a fast Postscript printer.

## EXAMPLES

```
xsat -if=vis -cu=la -de=d
```

This displays the latest visible satellite image. For most images, this will only be a subset of the entire image. To get the full image, either zoom or remap needs to be done.

```
xsat -if=vis -cu=la -pd=us -de=d
```

This will remap the image to a US plot domain.

```
xsat -if=ir -cu=la -cof=sat_mb.enh -de=d
```

This will enhance the latest IR image with the MB enhancement scheme.

```
xsat -if=vis -cu=la -pd=us -pl=none -of=img
```

This remaps the image and saves the output into an image file for later use.

## FILES

## SEE ALSO