

GEMPAK Scripting Exercise

For this workshop I have provided a set of scripts that each of you can download from github.com using git clone.

```
[awips@edex ~]$ git clone
https://github.com/Unidata/unidata-gempak-workshop.git
```

```
[awips@edex ~]$ cd unidata-gempak-workshop
[awips@edex ~]$ ls
alias.html  gdinv-files.sh  gdinv.sh  gdplot2.csh  grid_names.csv  level3
level3.csh  nidsid.tbl  prcp.nts  tables  tmpf.nts  trenberth.nts
```

The GDPLLOT2 script allows for any model and cycle input, and any forecast hour value (out to the maximum available for each model). Additionally, these .nts files present with this script are specified as a command line argument.

Note: This script is **not** idiot proof. This is simple shell scripting.

Running the following,

```
[awips@edex ~]$ ./gdplot2.csh 2014102700 15 HRRR tmpf.nts
```

will produce a series of images out to 15 hours for HRRR cycle 20141027 0000 UTC, with tmpf.nts as the input (below). The images will be written to the file `_${modelName}_${parm}`, such as “HRRR_tmpf” in the above example.

Inside the script, the inputs are taken to variables as

```
$0 = ./gdplot2.csh
$1 = 2014102700
$2 = 15
$3 = HRRR
$4 = tmpf.nts
```

The output device is defined in gdplot2.csh, and can be changed to XW to animate frames on the workstation, as long as gpend is also removed (or commented out) so that the XW session will not be

terminated after each frame as with dumping images.

tmpf.nts

```
GLEVEL  2          ! 10      ! 2
GVCORD  hght       ! hght
GDPFUN  tmpf       ! wnd      ! tmpf
TYPE    cf         ! b        ! c
CONTUR  3/3       ! 1        ! 1
CINT    1/32/32   !          ! 80
LINE    1/1/3     !          ! 1/1/2
FINT
-25;-20;-15;-10;-5;0;5;10;15;20;25;30;35;40;45;50;55;60;65;70;75;80;85;90;95;1
00;105;110;115
FLINE   30-7
FLINE   2-25
WIND                    ! 32/0.7/1/112
CLRBAR  31/h/lc/.5;0/1;.018/|.8 !
TEXT    1/22/2/hw
PANEL   0
SCALE   0
SKIP    0
HILO
HLSYM
REFVEC
CLEAR   yes
MSCALE
STNPLT
LUTFIL  none
STREAM
POSN    4
COLORS  2
MARKER  2
IJSKIP  4
GRDLBL  5
FILTER  yes
SKIP    0/2;2
```

prcp.nts

```
GLEVEL  0          ! 0          ! 0
GVCORD  none       ! none       ! none
GDPFUN  pmsl       ! p01i       ! mask(sle(wxtr,0),sge(p01i,.01))
TYPE    C          ! f          ! f
```

```

CINT      2/900/1200  !           !
LINE      8=155:155:155///3      ! !
FINT      !   .01;.05;.1;.15;.2;.25;.3;.35;.4;.45;.5;.55;.6;.65;.7;.75;1;1.25 !
/.1;.1
FLINE     !   0;21-30;14-20;5 ! 0;31/7
PANEL     0
SKIP      0/2;2
SCALE     0
CONTUR    10      ! 3/0 !
HILO      0      ! 32;0/X#2;N/.1-100/10/20;0/
HLSYM     1.3     ! 2;1/2/32/2 !
CLRBAR    31/h/lc/.5;0/1;.018/|.8 ! 0
WIND      ak7/0.45/1//0.01
REFVEC
TEXT      0.7/22/1/hw
CLEAR     y
GAREA     grid
IJSKIP    4

```

trenberth.nts

```

GLEVEL    700           ! 500:1000 ! 500:1000
GVCORD    pres
PANEL     0
SKIP
SCALE     5           ! 0 ! 0
GDPFUN    avor(wnd)    ! ldf(hght) ! thrm(hght)
TYPE      f           ! c ! A
CONTUR    3
CINT      ! 60
LINE      ! 31//3
FINT      2/-4/30
FLINE     28;28-13-1
HILO
HLSYM
CLRBAR    31/h/lc/.5;0/1;.018/|.8 !
WIND      0 ! ! am31/0.2//0.5
REFVEC    10
TITLE     1/-1       ! 1/-2 ! 1/-3
TEXT      0.65/2//hw
CLEAR     n
IJSKIP    10 ! ! 40
MAP       1
MSCALE
LATLON    0

```

SCRIPT TEMPLATES

A basic script template with looping

```
@ HourCount = 0
while ( $HourCount <= $2 )

    # do stuff here

@ HourCount = $HourCount + ${inc}
end
```

Where \$2 is the final forecast hour (15 in our example), and \$inc is the increment hour value, determined by the model name (1 hr increment for some models like HRRR, 3 hr for others).

To run a program such as gdplot2 inside such a loop,

```
@ HourCount = 0
while ( $HourCount <= $2 )

    gdplot2 << EOF

        # define stuff here

    r

    e
    EOF
    gpend # <- remeber this if DEVICE = gif

@ HourCount = $HourCount + ${inc}
end

scp file.gif server:/somewhere/else/
exit
```