PROGRAMMING "CONVENTIONS"

- Use APPS_DEFAULTS when possible ***INSTEAD*** of hardwiring directory paths in code.

- Program in the language you are comfortable with (FORTRAN, C, C++). Use shell scripts when appropriate (Perl, C shell, POSIX).

- Use the appropriate ***Documentation Block*** in each script or routine.

- Provide a lot of in-line documentation (via ***comment lines***) in the code itself! (You will thank yourself later, and so will anyone else who has to modify the code!!)

- Code goes in the **/rfc_arc/src/** directory on the development box.

- Follow the Application Integration Framework Manual (AIFM) where possible.
  www.nws.noaa.gov/tdl/awips

C-language Documentation Block

```
/* ROUTINE:
*
*  PURPOSE:
*
*  CALLED BY:
*
*  DATA SET(S) CALLED:
*
*  VARIABLES:
*
*  NON-SYSTEM ROUTINES USED:
*
*  NOTES:
*
*  REVISION HISTORY:
*    DATE    WHO/OFFICE          WHAT
*
*  RUN STRING:/CALLING SEQUENCE:
*
*****************************************************************/
#include “xxx.h”
...

void routine1();     /* linked routine */
int routine2();
void routine3();     /* linked routine */
...

Xxxxxxx()
{
....
}
```

C Example: Main Routine

```
/* ROUTINE:  main.c (main routine for HydDsply program)
*
*  PURPOSE:  to display archived observational data
*
*  CALLED BY:  N/A
*
```

```
*  DATA SET(S) USED:
*    netCDF historic data files (input/output)
*    User inputs (“requests”) (input)
*
*  VARIABLES:
*    *fin - (FILE pointer) for files read in (defaults, etc.)
*    datetime - (dttm struct) holds date/time info
*    ud - (userdefaults struct) holds user default settings
*    ...
*
*  NON-SYSTEM ROUTINES CALLED:
*    assign_parsed, check_group, ...
*
*  NOTES:  SQN_FILE is not deleted until the program ends - this
*            ensures that the same sequence number is used through the
*            entire run of the program.
*          Years are in 4-digit format.
*          ...
*
*  REVISION HISTORY:
*    DATE    WHO/OFFICE          WHAT
*    10/98   Lunsford/APRFC      Initial coding
*     2/99   Armour/APRFC        Added “f” (forecast) mode
*   ...
*
*  RUN STRING:
*    ./HydDsply
*
*********************************************************************/
#include “HydDsply.h”
...

void assign_parsed();        /* linked routine */
int check_group();
...

main()
{
...
}
```

C Example: Subroutine

```
/* ROUTINE: assign_parsed.c
*
*  PURPOSE:  to assign words parsed from input line into the proper
*            commands
*
*  CALLED BY:  main.c
*
*  DATA SET(S) USED:
*    parsed input line from user (input)
*
*  VARIABLES:
*    inputv[][INSTRLN] - (extern char) array of words from input line
*    ...
*
*  NON-SYSTEM ROUTINES CALLED:
```

```
*    check_group, determine_mode, process_flags, ...
*
*  NOTES: Only inputv[0] should have any upper case characters.
*
*  REVISION HISTORY:
*    DATE    WHO/OFFICE          WHAT
*    10/98   Armour/APRFC        Initial coding
*
*  CALLING SEQUENCE:
*    assign_parsed(sqn, *ninputs, station, &nsta, cur_year);
*  WHERE
*    sqn - (int) sequence number for this run
*    ninputs - (int) number of words the input line is parsed into
*    station - (string) array of station names
*    &nsta - (int pointer) number of stations in station array
*    cur_year - (int) 4-digit current year
*
*******************************************************************/
#include "HydDsply.h"
#include "HydDsplyShared.h"
...

int check_group();           /* linked routine */
void determine_mode();
...

void assign_parsed(int sqn, int *ninputs, char station[][STALN],
                   int *nsta, int cur_year)
{
....
}
```

<u>FORTRAN Documentation Block</u> (as in TDL attachment to AIFM)

```
C
      SUBROUTINE XXXXX(VAR1,...,VARN)
C
C     DATE WHO OFFICE COMPUTER
C
C     PURPOSE
C     THIS ROUTINE ...  CALLED BY XXX
C
C     DATA SET USE
C
C     VARIABLES
C
C     NON-SYSTEM ROUTINES CALLED
C
C     ************************************************************
C
C     DEFINE NUMBER OF VARIABLES
```

<u>FORTRAN Documentation Block</u> (alternate)

```
C
      SUBROUTINE XXXXX(VAR1,...,VARN)
C
C     PURPOSE
C
C     CALLED BY XXX
C
C     DATA SET USE:
C
C     VARIABLES:
C
C     NON-SYSTEM ROUTINES CALLED:
C
C     NOTES:
C
C     DATE WHO OFFICE COMPUTER
C
C     ************************************************************
C
C     DEFINE NUMBER OF VARIABLES
```

FORTRAN Example

```
C
      SUBROUTINE RDMETAR(CFILE,NREC,wmoId,stationName,latitude,
 1      longitude,elevation,timeObs,timeNominal,reportType,
 2      autoStationType,skyCover,skyLayerBase,visibility,
 ...    ...
 7      pressChange3Hour,correction,rawMETAR,NCODE)
C
C     MAY 1997 PEACHEY GSC HP
C     JULY 1999 MORRIS GSC HP
C     CHANGED pressChange3hour TO REAL FROM INTEGER
C
C     PURPOSE:
C     THIS ROUTINE RETURNS ALL THE VARIABLE VALUES IN A
C     METAR NETCDF FILE GIVEN THE FILE NAME.
C
C     DATA SET USE:
C     CFILE - NAME OF METAR NETCDF FILE (INPUT)
C
C     VARIABLES:
C     CFILE = METAR NETCDF FILE TO PROCESS (INPUT)
C       (CHARACTER*80)
C     NREC = NUMBER OF RECORDS IN THE FILE (INPUT)
C     wmoId(NREC) = WMO NUMERIC STATION ID (OUTPUT)
...   ...
C     NVDIMS(10) = VECTOR OF NVDIM DIMENSION IDS CORRESPONDING
C     TO VARIABLE DIMENSIONS (INTERNAL)
C     NVS = NUMBER OF VARIABLE ATTRIBUTES (INTERNAL)
C
C     ROUTINES CALLED:
C     NCOPN, NCINQ, NCDINQ, NCVINQ, NCVID, NCVGT, NCVGTC
C
C     ************************************************************
C
C     DEFINE NUMBER OF VARIABLES
      PARAMETER (NVARS=32)
C
      CHARACTER*80 CFILE
      CHARACTER*1 stationName(5,NREC)
      ...
```

Script Documentation Block

```
/* NAME:  (include file extension, i.e., myScript.csh)
*
*  PURPOSE:
*
*  DATA SET(S) CALLED:
*
*  VARIABLES:
*
*  NOTES:
*
*  REVISION HISTORY:
*    DATE    WHO/OFFICE          WHAT
*
*  RUN STRING:
*
*****************************************************************/
```