

IX.5.5 FORECAST SYSTEM PROCESSED DATA BASE ACCESS

Introduction

The Processed Data Base (PDB) [[Hyperlink](#)] contains time series data produced by the Preprocessor Functions and the Forecast Component.

Access Routines

A description of the routines that can be used to access the PDB can be found in Section IX.3.5 [[Hyperlink](#)].

Access Notes

See Section IX.5.1 [[Hyperlink](#)] for general access information.

Routine RPDBCI must be called before data can be read from the PDB. This routine read control information from the PDB into common blocks used by the read/write routines.

Routine WPDBCO must be called after data has been written to the PDB. This routine write control information from common blocks used by the read/write routines to the PDB. WPDBCO must be called if any control information has changed.

If data has been written to the PDB, all files must be closed by calling routine UCLOSL.

Example

The following example shows how to:

- o read a time series header
- o read time series data
- o write time series data

```
C-----  
C  
C      PROGRAM MAIN  
C  
C      CHARACTER*1 ANS  
C      CHARACTER*4 ZTIME/'Z'/  
C      CHARACTER*4 TSTYPE,TSUNIT,TSTYPH,TSUNITH  
C      CHARACTER*8 TSID,TSIDH,FTSID  
C      CHARACTER*20 TSDESC  
C      CHARACTER*72 CARD  
C      DIMENSION IHEAD(22)  
C      PARAMETER (LTSDAT=2000)  
C      DIMENSION TSDAT(LTSDAT)  
C      PARAMETER (LWKBUF=1000)  
C      DIMENSION IWKBUF(LWKBUF)  
C  
C      INCLUDE 'uiox'  
C      INCLUDE 'upagex'  
C
```

```

C
C OPEN INPUT/OUTPUT FILES
  CALL UPRIMO
C
C GET USER IDENTIFIER
  CALL HGTUSR (PUSRID, ISTAT)
C
C PRINT USER IDENTIFIER
  WRITE (LP, *) 'USER = ', PUSRID
C
C STORE CURRENT HYDROLOGIC DATES IN COMMON BLOCKS
  CALL HSYSDA (JULDAT)
C
C READ TIME SERIES IDENTIFIER, DATA TYPE AND DATA UNITS
10  TSID=' '
    TSTYPE=' '
    TSUNIT=' '
    WRITE (LP, *) 'ENTER IDENTIFIER, TYPE AND UNITS, ',
*      ''OFSTEST'' OR <RETURN> TO QUIT:'
    READ (ICD, ' (A)', END=50) CARD
    NSCAN=1
    CALL USCAN2 (CARD, ' ', NSCAN, TSID, LTSID, ISTAT)
    NSCAN=2
    CALL USCAN2 (CARD, ' ', NSCAN, TSTYPE, LTSTYPE, ISTAT)
    NSCAN=3
    CALL USCAN2 (CARD, ' ', NSCAN, TSUNIT, LTSUNIT, ISTAT)
C
C CONVERT TO UPPER CASE
  NCHAR=0
  NCONVT=0
  CALL ULC2CP (TSID, LTSID, NCHAR, NCONVT, ISTAT)
  CALL ULC2CP (TSTYPE, LTSTYPE, NCHAR, NCONVT, ISTAT)
  CALL ULC2CP (TSUNIT, LTSUNIT, NCHAR, NCONVT, ISTAT)
C
  IF (TSID.EQ.' ') GO TO 50
  IF (TSID.EQ.'OFSTEST') THEN
    WRITE (LP, *) 'IDENTIFIER ', TSID, ' SPECIFIED'
    TSID='DARNC'
    TSTYPE='MAP'
    TSUNIT='MM'
    WRITE (LP, *) 'TSID SET TO ', TSID,
*      ' TSTYPE SET TO ', TSTYPE,
*      ' AND ',
*      ' TSUNIT SET TO ', TSUNIT
  ELSE
    WRITE (LP, *) 'TSID=', TSID, ' TSTYPE=', TSTYPE,
*      ' TSUNIT=', TSUNIT
  ENDIF
C
C READ DATA BASE CONTROL RECORDS
  CALL RPDBCI (ISTAT)
  WRITE (LP, *) 'RPDBCI CALLED : ISTAT=', ISTAT
  IF (ISTAT.GT.0) GO TO 50
C
C READ TIME SERIES HEADER
  LXBUF=1
  CALL RPRDH (TSID, TSTYPE, LXBUF, IHEAD, NXBUF, XBUF, FTSID, ISTAT)
  WRITE (LP, *) 'RPRDH CALLED : ISTAT=', ISTAT
  IF (ISTAT.GT.0) GO TO 10
C
C PRINT TIME SERIES HEADER
CCC  WRITE (LP, ' (A,22A4) ') 'IHEAD IN A4 FORMAT = ', IHEAD
CCC  WRITE (LP, ' (A,22I4) ') 'IHEAD IN I4 FORMAT = ', IHEAD
    ITMINT=IHEAD(2)
    NVLINT=IHEAD(3)
    NTSMAX=IHEAD(4)
    NTSVAL=IHEAD(5)
    CALL UMEMOV (IHEAD(8), TSIDH, 2)

```

```

CALL UMEMOV (IHEAD(10),TSTYPEH,1)
CALL UMEMOV (IHEAD(11),TSUNITH,1)
CALL UMEMOV (IHEAD(12),TSLAT,1)
CALL UMEMOV (IHEAD(13),TSLON,1)
JHRBEG=IHEAD(14)
CALL UMEMOV (IHEAD(18),TSDDESC,5)
WRITE (LP,60)
WRITE (LP,70) TSIDH,TSLAT,TSLON,TSDDESC
WRITE (LP,80) ITMINT,TSTYPEH,TSUNITH
C
C DETERMINE DATE DATA BEGINS
  NHOUR=0
  CALL GETDAT (JHRBEG,NHOUR,ZTIME,NBMO,NBDA,NBYR,NBHR,NTZC,IDLS)
C
C DETERMINE DATE DATA ENDS
  NHOUR=NTSVAL/NVLINT*ITMINT-ITMINT
  CALL GETDAT (JHRBEG,NHOUR,ZTIME,NEMO,NEDA,NEYR,NEHR,NTZC,IDLS)
C
C PRINT TIME SERIES PERIOD OF RECORD
  WRITE (LP,90) NBMO,NBDA,NBYR,NBHR,ZTIME,NEMO,NEDA,NEYR,NEHR,
*   ZTIME
C
C READ TIME SERIES DATA
  NUMPD=NTSVAL/NVLINT
  RMISS=-999.
  IFPTR=0
  CALL RPRDD (TSID,TSTYPE,JHRBEG,ITMINT,NUMPD,TSUNIT,RMISS,
*   LTSDAT,TSDAT,IFPTR,LWKBUF,IWKBUF,ISTAT)
  WRITE (LP,*) 'RPRDD CALLED : ISTAT=',ISTAT
  IF (ISTAT.GT.0.AND.ISTAT.NE.2) GO TO 10
C
C PRINT TIME SERIES DATA
  WRITE (LP,100) TSUNIT
  WRITE (LP,110) (TSDAT(I),I=1,NUMPD)
C
  WRITE (LP,*) 'OKAY TO WRITE DATA TO TIME SERIES ',
*   '(<RETURN> FOR NO)?'
  READ (ICD,'(A)') ANS
  IF (ANS.EQ.' ') GO TO 10
  IF (ANS.NE.'Y'.AND.ANS.NE.'y') GO TO 10
C
C READ YEAR, JULIAN DAY AND Z-TIME HOUR FOR DATA TO BE WRITTEN
  WRITE (LP,*)
*   'ENTER YEAR, JULIAN DAY AND HOUR OF FIRST DATA VALUE ',
*   'OR <RETURN> TO ADD DATA TO END TIME PERIOD: '
  READ (ICD,'(I2,1X,I3,1X,I2)',END=20) IYEAR,JULDAY,IZHOUR
  GO TO 30
20  IYEAR=0
    JULDAY=0
    IZHOUR=0
30  IF (IYEAR.EQ.0) THEN
C    ADD DATA TO END
      JHOUR=JHRBEG
      IBEG=NTSVAL
    ELSE
      WRITE (LP,*) 'IYEAR=',IYEAR,' JULDAY=',JULDAY,
*   ' IZHOUR=',IZHOUR
      JHOUR=(IYEAR*365+IYEAR/4+JULDAY-1)*24+IZHOUR
      IBEG=0
    ENDIF
C
C WRITE TIME SERIES DATA
  NDAYS=2
  NUMPD=(24/ITMINT)*NDAYS
  NVALS=NUMPD*NVLINT
  DO 40 I=1,NVALS
    TSDAT (IBEG+I)=I*1.0
40  CONTINUE

```

```

        IF (IBEG.GT.0) THEN
            NUMPD=NTSVAL+NVALS
            NVALS=NUMPD*NVLINT
        ENDIF
        IFPTR=0
        ICALL=0
        IREC=0
        WRITE (LP,*) 'JHOUR=',JHOUR
        CALL WPRDD (TSID,TSTYPE,JHOUR,ITMINT,NUMPD,TSUNIT,NVALS,
*      LTSDAT,TSDAT,IFPTR,ICALL,LWKBUF,IWKBUF,IREC,ISTAT)
        WRITE (LP,*) 'WPRDD CALLED : ISTAT=',ISTAT
        IF (ISTAT.GT.0) GO TO 10
C
C WRITE CONTROL RECORDS TO FILE
        CALL WPDBCO (ISTAT)
        WRITE (LP,*) 'WPDBCO CALLED : ISTAT=',ISTAT
        IF (ISTAT.GT.0) GO TO 10
C
C CLOSE FILES
        CALL UCLOSL
C
C READ TIME SERIES HEADER
        CALL RPRDH (TSID,TSTYPE,LXBUF,IHEAD,NXBUF,XBUF,FTSID,ISTAT)
        WRITE (LP,*) 'RPRDH CALLED : ISTAT=',ISTAT
        IF (ISTAT.GT.0) GO TO 10
C
C PRINT TIME SERIES HEADER
        ITMINT=IHEAD(2)
        NVLINT=IHEAD(3)
        NTSMAX=IHEAD(4)
        NTSVAL=IHEAD(5)
        CALL UMEMOV (IHEAD(8),TSIDH,2)
        CALL UMEMOV (IHEAD(10),TSTYPEH,1)
        CALL UMEMOV (IHEAD(11),TSUNITH,1)
        CALL UMEMOV (IHEAD(12),TSLAT,1)
        CALL UMEMOV (IHEAD(13),TSLON,1)
        JHRBEG=IHEAD(14)
        CALL UMEMOV (IHEAD(18),TSDESC,5)
        WRITE (LP,60)
        WRITE (LP,70) TSIDH,TSLAT,TSLON,TSDESC
        WRITE (LP,80) ITMINT,TSTYPEH,TSUNITH
C
C DETERMINE DATE DATA BEGINS
        NHOURL=0
        CALL GETDATE (JHRBEG,NHOURL,ZTIME,NBMO,NBDA,NBYR,NBHR,NTZC,IDLS)
C
C DETERMINE DATE DATA ENDS
        NHOURL=NTSVAL/NVLINT*ITMINT-ITMINT
        CALL GETDATE (JHRBEG,NHOURL,ZTIME,NEMO,NEDA,NEYR,NEHR,NTZC,IDLS)
C
C PRINT TIME SERIES PERIOD OF RECORD
        WRITE (LP,90) NBMO,NBDA,NBYR,NBHR,ZTIME,NEMO,NEDA,NEYR,NEHR,
*      ZTIME
C
C READ TIME SERIES DATA
        RMISS=-999.
        IFPTR=0
        CALL RPRDD (TSID,TSTYPE,JHRBEG,ITMINT,NUMPD,TSUNIT,RMISS,
*      LTSDAT,TSDAT,IFPTR,LWKBUF,IWKBUF,ISTAT)
        WRITE (LP,*) 'RPRDD CALLED : ISTAT=',ISTAT
        IF (ISTAT.GT.0.AND.ISTAT.NE.2) GO TO 10
C
C PRINT TIME SERIES DATA
        WRITE (LP,100) TSUNIT
        WRITE (LP,110) (TSDAT(I),I=1,NUMPD)
        GO TO 10
C
50 WRITE (LP,*) '- PROCESSING COMPLETE -'

```

```

        STOP
C
60   FORMAT (' - TIME SERIES HEADER -')
70   FORMAT (' TIME SERIES ID = ',A,3X,
*     'LATITUDE = ',F6.2,3X,
*     'LONGITUDE = ',F6.2,3X,
*     'DESCRIPTION = ',A)
80   FORMAT (' TIME INTERVAL = ',I2.2,3X,
*     'DATA TYPE = ',A,3X,
*     'DATA UNITS = ',A)
90   FORMAT (' ',
*     'DATE OF FIRST DATA VALUE = ',
*     I2.2,'/',I2.2,'/',I4,'-',I2.2,A,3X,
*     'DATE OF LAST DATA VALUE = ',
*     I2.2,'/',I2.2,'/',I4,'-',I2.2,A)
100  FORMAT (' - TIME SERIES DATA IN UNITS OF ',A,' -')
110  FORMAT (' TSDAT=',10F10.3)
C
        END
C
C-----
C
        SUBROUTINE GETDATE (JHRBEG,NHOUR,ZTIME,NEMO,NEDA,NEYR,NEHR,
*     NTZC,IDLS)
C
C
        NHOPDB=12
        JHOUR=JHRBEG+NHOUR-NHOPDB
        JDAY=JHOUR/24
        IHOUR=JHOUR-JDAY*24
        IF (IHOUR.GT.0) JDAY=JDAY+1
        CALL MDYH2 (JDAY,IHOUR,NEMO,NEDA,NEYR,NEHR,NTZC,IDLS,ZTIME)
C
        RETURN
C
        END

```