

## ofsde Processing for Bld OB1

### **Overview**

ofsde processes real time data from the Informix database and outputs a file for use by the OFS batchpst process. Types of processing performed by ofsde include transforming PC data into PP values, changing two character SHEF pe types into OFS types, station id translation based on SHEF source code and checking end times against a window around 12z for PP24 data. It is submitted via the cron between 2 and 5 times per hour.

For more information about the batchpst process, see the documentation in the NWSRFS Users Manual Chap VI p 4.3-1.

### **New for Bld OB1**

- changed pc2pp logic to include only PC data for given type/source - previously included all PC data regardless of type/source - problem reported by APRFC
- when attempting to get record from OFSDatatTrans table - if record not found in table, then print message to log and do NOT process data record (previously, if record in table was not found, data record was processed and written to vl2v5.buf with OFS pe blank)

### **New for Bld 5.2.2**

- new functionality to change obs times within a window around 12z for observed RRS data
- new .Apps\_defaults token ofsde\_rrstime\_check to turn on above time check

### **New for Bld 5.2.1**

- executable moved to /awips/hydroapps/rfc/nwsrfs/ofsde/bin/RELEASE
- script moved to ../nwsrfs/ofs/scripts dir
- log files moved to ../nwsrfs/ofs/output/ofsde\_logs
- added print of operating system (Linux or HP-UX) to log
- Linux version of executable created

### **New for Bld 5.1.2**

- changed to store data from ofsdatatrans and ofsstntrans tables first time referenced and then reuse
- added check on shef\_procobs token before processing ProcValue table records
- changed processing of PC data to read through records twice - first time to count records for malloc, second time for storing date/time and value
- removed qc check on forecast data (qc check on this data now being done in shefdecode)
- shefdecode has been changed to insert a postingtime for each

record which is the actual time the record was posted to the database - previously, the same postingtime was used for all records in a SHEF product - this was causing some records to be missed when comparing postingtime with the time of last ofsde run

- a problem in RFCWide-MPE has been fixed which caused precip values updated from the MPE GUI to be written to the curprecip table with incorrect units

#### **New for Bld 5.1.1**

- replaced use of views for retrieving data with a search in IngestFilter table resulting in much faster retrieval times
- removed update of processed\_code field
- .Apps\_defaults token ofsde\_output\_dir renamed ofsde\_log\_dir

#### **New for Bld 5.0**

- new pe table, GateDam added
- replaced LocRangeCheck and DefaultRangeCheck tables with LocDataLimits and DataLimits tables

#### **Outline**

- (1) Open log file and vl2v5.buf file
- (2) Read parameters from .Apps\_defaults file
- (3) Open database
- (4) Set isolation level to dirty read
- (6) Query the PerfLog table to find the last ofsde run time
- (7) Get all records from CurPrecip table in linked list form with posting time later than last ofsde run time (records are ordered by pe,lid,datetime)

For all records found:

- (a) Check IngestFilter table for ofs\_input value
- (b) If ofs\_input = "T" then

- (c) If record has same id as previous record AND obstime is within same 12z - 12z period as previous record AND pe="PC", then

- (d) Take next record

Else

- (e) Get OFS data type and forward/backward window from the OFSDataTrans table

- (f) Process record

End if

End If

(8) Get all records from Temperature table in linked list form with posting time later than last ofsde run time

For all records found:

(a) Check IngestFilter table for ofs\_input value

(b) If ofs\_input = "T" then

(c) Get OFS data type from OFSDataTrans table

(d) Process record

End if

(9) Get all records from Height table in linked list form with posting time later than last ofsde run time (same for Discharge, Snow, Ice, Moisture, Lake, Ground, GateDam tables)

For all records found:

(a) Check IngestFilter table for ofs\_input value

(b) If ofs\_input = "T" then

(c) Get OFS data type from OFSDataTrans table

(d) Process record

End if

(10) Get all records from Evaporation table in linked list form with posting time later than last ofsde run time (same for Radiation, Agricultural, YUnique tables)

For all records found:

(a) Check IngestFilter table for ofs\_input value

(b) If ofs\_input = "T" then

(c) Get OFS data type from OFSDataTrans table

(d) Process record

End if

(11) Get all records from ProcValue table in linked list form with posting time later than last ofsde run time (records in the ProcValue table have SHEF type ="P")

For all records found:

(a) Check IngestFilter table for ofs\_input value

(b) If ofs\_input = "T" then

(c) Get OFS data type from OFSDataTrans table

(d) Process record

End if

(12) Get all records from FcstDischarge table in linked list form with posting time later than last ofsde run time

For all records found:

(a) Check IngestFilter table for ofs\_input value

(b) If ofs\_input = "T" then

(c) If record has same id,pe,validtime as previous record, then read next record

(d) Get OFS data type from OFSDataTrans table

(e) Process record

End if

(13) Get all records from FcstHeight table in linked list form with posting time later than last ofsde run time

For all records found:

(a) Check IngestFilter table for ofs\_input value

(b) If ofs\_input = "T" then

(c) If record has same id,pe,validtime as previous record, then read next record

(d) Get OFS data type from OFSDataTrans table

(e) Process record

End if

(14) Get all records from FcstTemp table in linked list form with posting time later than last ofsde run time

For all records found:

(a) Check IngestFilter table for ofs\_input value

(b) If ofs\_input = "T" then

(c) If record has same id,extremum,validtime as previous record, then read next record

(d) Process record

End if

(15) Write number of records processed, time of run to Perflog table

(16) Close database and exit

## Notes

- (1) No processing is done for records in the FcstPrecip, FcstOther, FishCount, Power, and WaterQuality tables.
- (2) List of pe tables:
  - Precip
  - CurPrecip
  - Temperature
  - Height
  - Discharge
  - Snow
  - Ice
  - Moisture
  - Lake
  - Ground
  - Radiation
  - Evaporation
  - Agricultural
  - YUnique
  - Weather
  - Wind
  - FcstDischarge
  - FcstHeight
  - FcstTemp
  - FcstPrecip
  - FcstOther
  - GateDam
  - FishCount
  - Power
  - WaterQuality

## Subprocesses

- (1) qcrng
  - range checks on data using Informix DataLimits and LocDataLimits tables
  - if station id does not appear in the LocDataLimits table AND the pe type does not appear in the DataLimits table, then data passes qc check
  - used only for PP data generated from PC data
- (2) pc2pp
  - calculate PP06 and PP24 data at sub synoptic times
  - call qcrng
  - call wr2ofs
- (3) mape
  - calculate TA24, US24, RC24 and TD24 data
  - call wr2ofs

- (4) wr2ofs
  - station identifier translation using OFSStationTrans table
  - PE translation from SHEF to V5 form using OFSDataTrans table
  - write records to vl2v5.buf file
- (5) rrs2ofs
  - same as wr2ofs except writes duration code and 'FUT' for future data
- (6) durdec
  - decodes the SHEF duration code into hours
- (7) chktatm
  - checks obs time of observed TA data for possible change to 12z
- (8) chkpostpp
  - checks obs time of observed PP data for possible posting to OFS and possible change of obs time to 12z
- (9) process observed precip records
  - for PC data:
    - call pc2pp
  - for PP data:
    - if qc flag (quality\_code field) from shefdecode = passed, then
      - if duration = 2001 or 1024 or 5004 then call chkpostpp
      - call wr2ofs
    - end if
- (10) process observed height, discharge, snow, ice, moisture, lake, ground (RRS data) records
  - if ofsde\_rrstime\_check = ON AND
    - obs time is not 12z AND
    - forward window and back window not both = 0.0 then
    - call chkrstime
  - call durdec
  - call rrs2ofs
- (11) process observed temperature records
  - if TD data for 12z found, then call mape
  - for max/min temperatures, check that min < max and max > min
  - call chktatm
  - call wr2ofs
- (12) process observed radiation, evaporation, agricultural, unique records and processed data (from ProcValue table)

- call wr2ofs
- (13) process forecast discharge records
  - call durdec
  - call rrs2ofs
- (14) process forecast height records
  - call durdec
  - call rrs2ofs
- (15) process forecast temperature records
  - PE translation from SHEF to V5 form using OFSDataTrans table
  - call wr2ofs

### **qcrng Process**

- (1) Check the data value against the max and min values from the LocDataLimits table for the station id, pe type and date
- (2) if value is between max and min then return status=passed qc test  
 else if value is outside of range, then return status=failed qc test  
  
 else if no record is found in the table then check value against max/min values from the DataLimits table for pe type and date
- (3) if value is between max and min then return status=passed qc test  
 else if value is outside of range, then return status=failed qc test
- (4) if no record is found in either table, then return status=passed qc test

### **pc2pp Process**

- (1) Read PC values for given id and ts for 12z to 12z period of interest (subroutine rd12bf) (see notes 1,4,5)
- (2) Find the 5 data values with obs date/time within allowable window around the sub-synoptic times  
 If no value is available within allowable window, then mark PC value as missing
- (3) Attempt to estimate missing sub-synoptic PC values (subroutine estpc)

```

If a PC value is missing, then
  (a) find closest earlier and closest later PC values
      which bracket (in time) the missing PC value
  (b) if either value is not available, then go to (4)
  (c) if the two values are within 0.05 in absolute value,
      then set the missing value to the closest later
      PC value
End If

(4) Check Informix Ingestfilter table for PP duration = 1006
    If not found, then go to (10)

(5) Calculate 6 hour period totals

(6) Check CurPrecip table for a record with the same id, obs time
    with SHEF type/source = RG
    (A) If found, then go to step (9) - do not write data to
        vl2v5.buf

(7) Execute subroutine qcrng
    (A) If data fails qc test, then set value = -999.

(8) Execute subroutine wr2ofs

(9) Repeat steps (6), (7) and (8) for the three remaining periods

(10) Calculate 12z-12z total or partial day total
    For partial day total:
    if the PC value for the earlier 12z time is missing, then
        PP24 value is not calculated
    else
        if number of additional sub-synoptic PC values is > 0
        then
            if the latest sub-synoptic time of the PC value is
            > 18z, then
                partial day total = PC(latest obs time) - PC(previous
                12z)
            else
                PP24 value set to missing
        else
            PP24 value set to missing

(11) Check CurPrecip table for a record with the same id, obs
    time with SHEF type/source = RG
    (A) If found, then go to step (14) - do not write data to
        vl2v5.buf

(12) Execute subroutine qcrng
    (A) If data fails qc test, then set value = -999.

```



(13) Execute subroutine wr2ofs

(14) Exit

\*\*\*\*\*

#### Notes

- (1) Step (1) also saves PC values before the earlier 12z time and after the later 12z time for use in estimating missing PC values in step (3).
- (2) PP06 values are assigned times = 000000,60000,120000,180000.
- (3) The allowable window on obs time is read from the OFSDataTrans table. Separate values for forward in time and back in time windows are stored.
- (4) Step (1) reads through the PC records in the 12z - 12z time period twice. The first time to count the number of records for a malloc and the second time to actually store the values. To accomplish this, records are selected from the database ONCE, placed into a linked list and the list is traversed twice.
- (5) Prior to Bld OB1, all available PC data for the given id was used regardless of ts.

#### mape Process

- (1) Read latitude from Informix Location table for this station (latitude is used by RC24 calculation)
- (2) Choose beginning/ending obs dates of 12z-12z period as in pc2pp process
- (3) For data types = TA,TD,US,XC
- (4) Read all available data values for 12z-12z period of interest with duration code = 0 (subroutine rd1212)
- (5) If there are at least 8 values available and at least 4 values for each date, then set the 24 hour value calculation flag  
Else 24 hour value calculation flag = clear  
End If
- (6) If value > -25. and value < 200. , then retain value
- (7) If less than 8 values are both non missing and not out of range, then 24 hour value calculation flag = clear
- (8) If data type = TA then  
If 24 hour value calculation flag = set, then calculate the average  
Else if TAIRZN and TAIRZX are both available from the

```
Temperature table for date of interest,  
then calculate average of TAIRZN and TAIRZX values  
Else set average to missing  
End If
```

```
Else if data type = US or TD then  
If 24 hour value calculation flag = set, then calculate  
average  
Else set average to missing  
End If
```

```
Else if data type = XC then  
Based on latitude of station and month, calculate number  
of hours of daylight expected  
Combine with non missing XC values from daylight hours  
to calculate RC24  
End If
```

(9) Execute subroutine wr2ofs

(10) Go to (3)

#### Notes

(1) XC data type is transformed into RC24.

TA " " " " " TA24.

TD " " " " " TD24.

US " " " " " US24.

RC24, TA24, TD24, US24 data types are stored as EA24 in the  
PPDB (see Section IX.3.4B-RPDDLY in NWSRFS Manual)

(2) XC = sky cover

US = wind speed

TD = dew point temperature

TA = air temperature

#### **Processing Forecast Temperature Data**

ndate = value of .Apps\_defaults token ofsde\_ndate  
if not defined, default value = 7 is used

maxfuptime = current time + ndate days

(1) search the FcstTemp table for records with validtime >  
current time AND validtime < maxfuptime

(2) If found, then

(A) Select records with newest basistime for each id, pe,  
ts, ex, validtime

(B) call rrs2ofs

End If

### **Processing Forecast Height and Discharge Data**

In step (1) below, the height data is selected from the fcstheight table while the discharge data is read from the fcstdischarge table.

(1) Search the table for records with validtime > current time

(2) If found, then

(A) Select records with newest basistime for each id, pe, ts, validtime

(B) call durdec

(C) If SHEF duration code is successfully transformed into hours, then call rrs2ofs

End If

(3) Exit

### **CHKPOSTPP**

This process checks the obs time and duration code of PP data to determine if the data should be posted to OFS. It also checks the obs time of the PP data being posted to OFS to determine if a change to the obs time is necessary. See documentation in NWSRFS Users Manual Chap VI p 4.2-5. This check is done for duration codes = 1024 and 2001 only.

iotime = observation time (hhmmss) (z time)

idur = SHEF duration code

intlppp, intuppp = .Apps\_defaults tokens

default values: intlppp = 2

intuppp = 2

iobl = 120000 - (intlppp\*10000)

iobu = 120000 + (intuppp\*10000)

if(iotime < iobl OR iotime > iobu) AND (idur = 2001 OR idur = 1024)

```

    set flag to not post data to OFS and return
else
    set flag to post data to OFS
end if

if(iotime >= iobl AND iotime <= iobu)

    iotime changed to 120000

end if

```

### **CHKTATM Process**

This process checks the obs time of max/min temperatures to determine if a change to the obs time is necessary. This is done to satisfy the requirements of OFS. See documentation in NWSRFS Users Manual Chap VI p 4.2-5.

```

shs = SHEF source code
shex = SHEF extremum code
iotime = observation time (hhmmss) (z time)
intlXXX, intuXXX = .Apps_defaults tokens

```

```

default values: intlrmn = 8
                 inturmn = 2
                 intlrzn = 2
                 inturzn = 2
                 intlrxz = 8
                 inturzx = 2

```

```

if shs = M then

    if shex = N then

        iobl = 12 - intlrmn
        iobu = 12 + inturmn

        if(iotime >= iobl AND iotime <= iobu) then iotime changed
            to 12z

    else if shex = X then

        iobl = 12 - intlrxz
        iobu = 12 + inturzx

        if(iotime >= iobl AND iotime <= iobu) then iotime changed

```

```

        to 12z

    end if

else if shs = Z then

    if shex = N OR shex = P then

        iobl = 12 - intlrrzn
        iobu = 12 + inturzn

        if(iotime >= iobl AND iotime <= iobu) then iotime changed
            to 12z

    else if shex = X then

        iobl = 12 - intlrrzx
        iobu = 12 + inturzx

```

### **CHKRRSTIME Process**

This process was added in Bld 5.2.2 at the request of NWRFC to check obs times of observed RRS data against a window around 12z. Originally, NWRFC requested this time check for Discharge data only. During testing, it was then decided to generalize it for all RRS data types. If the obs time is not 12z but is within the window, then the obs time is changed to 12z before the record is passed to batchpst. The window is defined by reading the `fwd_time_window` and `bkw_time_window` fields from the `OFSDDataTrans` table. Units of these values is hours.

```

iotime = observation time (hhmmss) (z time)
ifwin = integer portion of forward window
ibwin = integer portion of backward window

```

```

idfwin = decimal portion of forward window
idbwin = decimal portion of backward window

```

```

iobjf= 120000 + (ifwin*10000) + (idfwin*6000)
iobjb= 120000 - (ibwin*10000) - (idbwin*6000)

```

```

if (iotime >= iobjb AND iotime <= iobjf) then change iotime to 12z

```

### **DURDEC Process**

The durdec process decodes the duration code into a duration in hours and in some cases changes the hour of observation. It is

used by both siipp and ofsde.

The following is an outline describing the procedure:

Input: SHEF duration code, hour of observation (=hrobs), windows around observation time (=intlppp and intuppp)

Output: duration in hours (=durhr), hour of observation (=hrobs)

- (1) If SHEF duration code = 0, then
  - (a) durhr = 0
  - (b) hrobs unchanged
  - (c) returnEnd If
  
- (2) Decode duration code into first digit (=type) and last two digits (=nhrs)
  
- (3) If type = 1, then
  - (a) durhr = nhrs
  - (b) hrobs unchangedElse If type = 2, then
  - (c) durhr = (nhrs \* 24)
  - (d) if hrobs >= 12, then
    - (i) iw = hrobs - 12
    - (ii) int = intlpppelse
    - (iii) iw = 12 - hrobs
    - (iv) int = intupppend if
  - (e) if iw <= int, then hrobs = 12Else If type = 5, then
  - (f) if hrobs >= 12, then
    - (i) iw = hrobs - 12
    - (ii) int = intlpppelse
    - (iii) iw = 12 - hrobs
    - (iv) int = intupppend if
  - (g) if iw <= int, then
    - (i) durhr = 24
    - (ii) hrobs = 12else
    - (iii) durhr = hrobs - 12
    - (iv) if durhr < 0, then durhr = hrobs + 12end ifElse SHEF duration code cannot be transformed into hours  
End If

Notes

(1) intlppp and intuppp are currently read from .Apps\_defaults.

### **vl2v5.buf File Format**

The vl2v5.buf is an ascii file which is output from ofsde and read by batchpst.

<u>Field Description</u>	<u>Format</u>
station identifier	A8
date (yyyymmdd)	I8
time (hhmm)	I4.4
PE code for OFS	A4
value	F11.2

Following fields are written for RRS data only:

future data indicator	A3
duration (hours)	I2

### Notes

- (1) There is no space between the date and time fields. All other fields are separated by one space.
- (2) Values greater than 1e8 are checked for and flagged as errors. This prevents the case of the value field containing \*\*\*\*\* as is the case when the format is exceeded.
- (3) The future data indicator field = "FUT" for future RRS data and blank for observed RRS data.
- (4) The duration is zero for instantaneous data.

### **.Apps\_defaults tokens**

```
db_name           : hd5_22XXX           # database name
server_name       : ONLINE              # server name
ofs_scripts      : $(ofs_dir)/scripts
ofsde_log_dir    : $(ofs_output)/ofsde_logs # dir containing log
```

```

# files and vl2v5.buf
# file

intlppp      : 2      # number of hours before 12z for
# changing ppp obstime to 12z
# number of hours before 12z for
# posting 1024 and 2001 data to OFS

intuppp      : 2      # number of hours after 12z for
# changing ppp obstime to 12z
# number of hours after 12z for posting
# 1024 and 2001 data to OFS

ofsde_ndate  : 7      # number of days to search for forecast
# temperature data

ofsde_rrstime_check : ON # check obs time of observed RRS
# data against window around 12z

# intervals for max/min temperatures
# these represent number of hours around 12z
# naming scheme: intXYZ
#
# X = l - lower interval
# u - upper interval
#
# YY = SHEF type/source = RM or RZ
#
# Z = SHEF extremum code = X or N

intlrmn      : 8
inturmn      : 2
intlrmn      : 2
inturzn      : 2
intlrmn      : 8
inturzn      : 2

```

### **ofsde Files**

(1) Script for running ofsde from cron:

```
/awips/hydroapps/rfc/nwsrfs/ofs/scripts/run_ofsde
```

(Combines execution of ofsde and batchpst)

(2) ofsde executable:

```
/aw*/hy*/rfc/nwsrfs/ofsde/bin/RELEASE/ofsde
```



(3) ofsde log files:

```
/aw*/hy*/rfc/nwsrfs/ofs/output/ofsde_logs/ofsde.mmddyyyymm
```

(4) ofsde data input to batchpst:

```
/aw*/hy*/rfc/nwsrfs/ofs/output/ofsde_logs/vl2v5.buf
```

### **Interaction with MPE**

When a gage value is edited through the MPE GUI, the corresponding record in the curprecip (and precip) tables will have the value field updated to the new value and postingtime field updated to the current time. The postingtime field update will cause the record to be selected for processing during the next ofsde run.