

## F-1. shef\_decode\_raw

### 1.0 General Information

The primary method of getting data into the RFC Archive database, particularly raw data, is via SHEF messages decoded by the `shef_decode_raw` application. This program is run continuously in the background by the user `oper`, and is fed SHEF products in parallel with the IHFS database's SHEF decoder on the AWIPS dx1 system. Additionally, SHEF messages from `metar`, `ldad`, and WAN sources are fed into the archive system by placing them in the same queue as the SBN SHEF messages. The application is written in Fortran and `ecpg/C`.

The application writes to the following SHEF data value tables: *pedrsep*, *pecrsep*, *pemrsep*, *pedfsep*, *pehfsep*, *peqfsep*, *pedcsep*, *peoossep*, *unkstnvalue*, and *pairedvalues*. Only the data value table *commentvalue* is currently not written to. It is still on the future enhancements list for the `shef_decode_raw` application.

### 1.1 Design Considerations

There were numerous reported problems with the version 1 release of the archive SHEF decoders. After reviewing the problems and possible solutions at the Nov 17-18, 2003 RAXUM team meeting, the team recommended that the archive SHEF decoders be rewritten using the national operational (IHFS) SHEF decoder as the starting point. A natural product of this change would be log output which is more consistent with that produced by the IHFS decoder, a feature which many RFCs desired.

This recommendation was reviewed by OHD management in January 2004. A "re-design" review conference call in early February produced the following decisions:

- there will continue to be two RFC archive SHEF decoders (raw and processed),
- both decoders will use the same parser as the IHFS SHEF decoder,
- the format of the log files (daily and product) were changed to be similar to the log files of the IHFS SHEF decoder, and
- the incoming directory for the `shef_decode_raw` will remain on the dx1 and be mounted to the rax, while the `shef_decode_processed` incoming directory will remain on the rax with no mount back to the dx (formerly ds).

### 1.2 Enhancements/Bug Fixes/Changes

#### **Build OB8.3**

##### ***Enhancements***

- DR#18810 Inclusion of tidal data and other shorter time step data (<15 minutes) into the RFC Archive database and to make this data available to the verification

software. This enhancement creates a new database table called *peirsep* and changes were made to the *shef\_decode\_raw* and verification software. One new apps\_defaults token, *adb\_shef\_raw\_storall*, was created as a result of this enhancement.

### **Bug Fixes**

- DR#18798 The *shef\_decode\_raw* application is incorrectly testing data limits.

### **Documentation**

- Sections 1.2, 2.1 and 4.0 have been updated.

### **Build OB7.2**

The application was converted to *ecpg/C* from *esql/C* (Informix) to work with Postgres RDMBS. Documentation was updated to reflect this change.

### **Enhancements**

- two new apps\_defaults tokens, these are:
  - adb\_shef\_raw\_tmp\_dir* :  $\$(adb\_raw\_que)$
  - adb\_shef\_raw\_add\_adjust* : OFF
- use of the *adjustfactor* table enabled

### **Build OB6**

### **Documentation**

Section 1.2 has been updated.

### **Enhancements**

- None.

### **Bug Fixes**

- Implemented table changes to *unkstnvalue* and *pehfsep* for OB6. Specifically, the "quality\_code" field has been dropped from *unkstnvalue*, and the fields "quality\_code", "revision", "product\_id", "producttime" and "postingtime" have been dropped from *pehfsep*. [HSD Bug r26-17]
- R1-52. When processing archive data (5000 values processed) the *shefdecoder* will crash when writing to the *pecrsep* table. In addition, the *shefdecoder* has also died when running the live data feed when writing to the *pecrsep* table. The interim release 1 of OB5 did not have this problem.
- R1-51. The raw decoder is doing some extra checks based on the SHEF data qualifier code, which shouldn't be occurring... it has been noticed particularly with HADS data. When the SHEF data qualifier code is certain values the *shefdecoder* throws the data away instead of posting it. The raw decoder should behave in a similar manner to the IHFS *shefdecoder*. [HSD Bug r26-29]
- R1-53. Dates with date-time prior to Jan 01, 1970 are not posting properly to the archive DB even though the *shefdecoders* posting summary information indicates

that the data posted. Some preliminary testing showed the current decoders appear to change any year that is before 1970 to 1970 when it posts the date. [HSD Bug r26-43]

- R1-54. The raw shefdecoder is not doing the datalimits/locdatalimits testing that it is supposed to be doing. When the shefdecoder processes a piece of data that should be flagged as "B" in the *pecrsep* table, it is being flagged as "G" instead. [HSD Bug r26-45]

## **Build OB5**

### **Documentation**

Sections 1.2, 2.1, 3.3, 4.0, 5.0 and 6.0 have been updated.

### **Enhancements**

- ER-9. Added user controlled posting error/warning messages capability. There are three new apps\_defaults tokens: *adb\_shef\_raw\_dupmess*, *adb\_shef\_raw\_locmess*, and *adb\_shef\_raw\_elgmess*.
- Added writing error messages to the log file whenever a query of the archive database fails unexpectedly.
- Added writing error messages to the *shef\_decode\_raw* log file when a value to post is outside the time window dictated by the tokens *adb\_shef\_winpast* and *adb\_shef\_winfuture*. The data is not posted in this case. This check is not applied to values with a type of 'F' (forecast) or 'C' (contingency). It is also not applied to values with a duration of 'M' (monthly).
- Added writing error messages to the *shef\_decode\_raw* log file when a value to post is a forecast (type of 'F') and has a basistime after the validtime. The data is still posted in this case.
- R1-19. The *shef\_decode\_raw* program has been enhanced to post data to the *pemrsep*, *peqfsep*, and *pehfsep* tables. For a SHEF record that is to be posted, if the d (duration) is 'M' and the t (type) is 'R', then the record is posted to the *pemrsep*. If the d is 'Q' and t is 'F', then it is posted to the *peqfsep* table. If the d is 'H' and the t is 'F', then it is posted to the *pehfsep* table. (Excludes *commentvalue* table)
- New apps\_defaults token *adb\_shef\_duplicate\_raw*
- R1-45. Updated the SHEFPARM file.

### **Bug Fixes**

- R1-23. For the *unkstnvalue* table, the SHEF decoder now computes the "idur" field of a data row correctly. (This bug could not be reproduced locally and we believe it was fixed before this release.)
- R1-24, R1-30. The SHEF decoder now correctly enforces the SHEF revision flag.
- R1-37. If the SHEF record to post is for forecast data (i.e. the t (type) in the *pedtsep* is 'F'), the decoders no longer check to see if the time of the record to post is within the window specified by the tokens *adb\_shef\_winfuture* and *adb\_shef\_winpast*. [HSD Bug r25-60]

- R1-39. Fixed an internal code problem that caused the decoders to crash in specific circumstances after processing only a couple records. This also corrected an unnumbered ADB bug involving a “fort.#####” file being created in the /rfc\_arc/bin directory that should not be created, where #'s are a number.
- R1-41. Fixed problem with insert/update counts in the summary information.
- R1-42. The SHEF decoder can now overwrite existing values in the database with missing values.
- Unnumbered Bug: Fixed the summation in the “Total” field of the log output. Previously, it was the total of ValIns, ValUpd, NoInsRec, and NoUpdRec. Now it is the sum of ValIns and ValUpd, so that it is the total number of values successfully posted to the database, either as an insert or update.
- Unnumbered Bug: The location and ingestfilter counters for the *unkstnvalue* table were changed in the log file so that they do not count values that were not posted due to the post flag of location being set to 0 or the ingest flag of ingestfilter being set to 0. Before, these values were included in the counters, even though no data was posted to *unkstnvalue*.
- Unnumbered Bug. Made the decoders more efficient by removing unneeded “trim” commands used within the SQL statements. Also, by correcting a problem with how the decoders dealt with database records in internal memory, the decoders now perform fewer insert/updates. Thanks to James Paul at ABRFC for identifying the solution to this problem.
- Unnumbered Bug. Corrected a problem involving the *datalimits* and *locdatalimits* tables not being used for gross and reasonable range checks. If the *locdatalimits* table provides limits for the lid and pedtsep of the current SHEF record, those limits are used. Otherwise, if the *datalimits* table provides limits for the pedtsep, those are used. In either case, if data limits are available and if either the gross or reasonable range limits are exceeded, the quality code is changed to reflect that.
- Unnumbered Bug. The performance logging feature was changed so that each decoder (raw and processed) uses a separate flag to turn it on. Specifically, the raw decoder uses the apps\_defaults token *adb\_shef\_raw\_perflog*, while the processed decoder uses the apps\_defaults token *adb\_shef\_pro\_perflog*. If on, the performance log will be provided in a file called *shef\_perf.log* in the logs directory for the decoder involved.

## **Build OB4**

### ***Enhancements***

See section 1.1 about the changes.

### ***Bug Fixes***

- r1-3 -- It has been reported that the *shef\_decode\_raw* program is running very slowly at two sites.
- r1-4 -- The raw decoder stopped working when a large number of *metar2shef* files (>400) were placed in the *ds /data/fxa/ispan/hydro\_adbs* directory.

- r1-6 -- Both the shef\_decode\_raw and shef\_decode\_pro programs end unexpectedly.

### Changes

- changes in apps\_defaults tokens
- eliminated the use of the nohup command in start script, nohup.out file is replaced with output being redirected into files raw.out and raw.err
- changed from using the inputparm file to the SHEFPARM file for the definitions of valid SHEF pedrsep codes
- eliminated the cfg file and and command line options feature
- the start and stop scripts were modified

## 2.0 Configuration Information

### 2.1 Apps\_defaults Tokens

The .Apps\_defaults and .Apps\_defaults\_site files are located in the /awips/hydroapps directory.

An excerpt from the .Apps\_defaults file follows. This list shows the tokens that are used by this SHEF decoder.

```
adb_name      : adb_ob7rha      # archive database name
adb_dir       : /rfc_arc       # Base RFC Archive Directory
adb_raw_que   : /rfc_arc_data/q/raw/ # pathname for raw q input directory
adb_bin_dir   : $(adb_dir)/bin  # pathname for the bin directory
adb_cfg_dir   : $(adb_dir)/cfg  # pathname for the config directory
adb_lib_dir   : $(adb_dir)/lib  # pathname for the lib directory
adb_logs_dir  : $(adb_dir)/logs # pathname for the logs directory
adb_scripts_dir: $(adb_dir)/scripts # pathname for the scripts directory
adb_shef_winpast : 10          # number of days in past to post data
adb_shef_winfuture : 30        # number of minutes in future to post obs data
shefdecode_rax_userid : oper    # controlling UNIX user
adb_shefdecode_input : $(adb_cfg_dir)/decoders # RAX SHEF parameter file
# location
adb_shef_raw_logs_dir : $(adb_logs_dir)/decoder/raw/logs # pathname for the
# daily logs directory
adb_shef_raw_err_dir : $(adb_logs_dir)/decoder/raw/err # pathname for the
# product logs directory
adb_shef_raw_keepperror : IF_ERROR # keep files (=ALWAYS) or only
# when errors occur (=IF_ERROR)
adb_shef_raw_post_unk : IDS_AND_DATA # NONE - do not post to the UnkStnValue table
# IDS_AND_DATA post to the UnkStnValue table
adb_shef_raw_checktab : ON       # ON checks location and ingestfilter tables
# to see if data should be posted
# OFF does not check location and
# ingestfilter tables before posting
adb_shef_raw_dupmess : ON       # ON/OFF
adb_shef_raw_locmess : ON       # ON/OFF
adb_shef_raw_elgmess : ON       # ON/OFF
adb_shef_duplicate_raw : USE_REVCODE # ALWAYS_OVERWRITE/USE_REVCODE
adb_shef_raw_perflog : OFF      # OFF/ON
```

---

```

adb_shef_raw_tmp_dir      :   $(adb_raw_que) # option to place files.list and SHEFOUT
                                #   files in directory other than the
                                #   incoming message directory (default)
adb_shef_raw_add_adjust   :   OFF           # OFF/ON (default is OFF)
                                # option to allow decoder to adjust data
                                # prior to posting
adb_shef_raw_storall      :   OFF           # OFF (default) writes only to the pecrsep
                                # table, ON writes to both the pecrsep
                                # and peirsep tables

```

### **The following tokens are new as of build OB5:**

```
adb_shef_raw_dupmess      : ON             # ON/OFF
```

If set to 'ON', a message will be written to the log file when a value to be posted would overwrite an existing value in the archive database (i.e., the value is a 'duplicate' - the numerical value may be different than what is in the database, but the pedtsep and time are the same). Messages will only be written if *adb\_shef\_duplicate\_raw* is set to 'USE\_REVCODE', the value to post is a 'duplicate' of an existing value, and there was no revision flag in the SHEF message. The message will state that the value was not posted due to its being a duplicate.

```
adb_shef_raw_locmess     : ON             # ON/OFF
```

If set to 'ON', a message will be written to the log file when a value to be posted is associated with a location (lid) that cannot be found in the *location* table of the archive database. The message will state that the value was posted to the *unkstnvalue* table if *adb\_shef\_raw\_post\_unk* is set to 'IDS\_AND\_DATA', or it will state that the value was discarded if *adb\_shef\_raw\_post\_unk* is set to 'NONE'.

```
adb_shef_raw_elgmess     : ON             # ON/OFF
```

If set to 'ON', a message will be written to the log file when a value to be posted is associated with a lid and pedtsep that is not found in the *ingestfilter* table. The message will state that the value was posted to the *unkstnvalue* table if *adb\_shef\_raw\_post\_unk* is set to 'IDS\_AND\_DATA', or it will state that the value was discarded if *adb\_shef\_raw\_post\_unk* is set to 'NONE'.

```
adb_shef_duplicate_raw   : USE_REVCODE #ALWAYS_OVERWRITE/USE_REVCODE
```

If the token is not found, it is assumed to be set to 'USE\_REVCODE'. When the token is set to 'USE\_REVCODE', the raw SHEF decoder will update duplicate data (i.e. a new value to post for which a value is already in the database) only when the SHEF revision flag is set. When the token is set to 'ALWAYS\_OVERWRITE', the raw SHEF decoder always updates duplicate data.

```
adb_shef_raw_perflog     : OFF           # OFF/ON
```

If set to 'ON', the performance log will be provided in a file called shef\_perf.log in the logs directory for the raw decoder.

**The following tokens are new in ob7.2:**

adb\_shef\_raw\_tmp\_dir : \$(adb\_raw\_que) # this is default setting

This token gives the RFC the option to place files.list and SHEFOUT files in a directory other than the incoming message directory. The default is the incoming message directory. This capability was added because of a problem that occurred during testing of the ob7.2 shef\_decode\_raw application where the SHEFOUT file was being purged by a process occasionally.

adb\_shef\_raw\_add\_adjust : OFF # OFF/ON (default is OFF)

This token gives the RFC the option to allow the shef\_decode\_raw application to adjust data prior to posting. If set to ON, any adjustment is performed as follows:

AFTER posting a value to the database, the decoder checks to see if the apps\_defaults token *adb\_shef\_raw\_add\_adjust* is not set to OFF. If it is not OFF, then the decoder will search the *adjustfactor* table rows for one to use. If it finds one (i.e. matching the primary key and the date associated with the adjust precedes the obstime of the value to post), then it will find the best one (the date precedes and is closest to the obstime) and apply the adjust. It will then post the adjusted record with a source of 9 (i.e. s = '9').

In order for the adjusted data to make it into the same table as the original data, an appropriate record must have previously been added by the RFC to the *ingestfilter* table with a source of '9'.

NOTE: Forecast data (i.e. data for which the type is 'F') is not affected, but all other types are.

**The following tokens are new in ob8.3:**

adb\_shef\_raw\_storall : OFF # OFF/ON (default is OFF)

As part of the DR#18810 small enhancement, a new table was added to the RFC Archive database's suite of SHEF data value tables. Its name is peirsep and its is a single value per row structure. There are some limitations, only the verification software, shef\_decode\_raw and adbpq.pl (forms data viewer/editor) *applications* can see this new table (peirsep). DatView cannot see this new table at this time. Another limitation is that since this is an ever growing table, when this new table gets really big, there are query performance issues. (Reference: *"Postgres Query Performance, Pseudo Array versus Single Value per Row Table Structure Test Results" that can be found on the RAXUM website, <http://www.nws.noaa.gov/oh/rfcdev/docs/testpseudoVSSingleFinal.pdf>.*

When the token is set to OFF (default) the 15-min to 6-hour time step data, Reading (observed) the `shef_decode_raw` application works just as it did in previous builds posting what data it can to the `pecrsep` table.

When the token is set to ON, the `shef_decode_raw` application write this data to both the `pecrsep` table and to the new `peirsep` table. Thus data with shorter time steps that gets lost when writing to the `pecrsep` step is always in the `peirsep` table, as it has no restrictions on the time step.

## 2.2 SHEFPARM File

This decoder uses the same SHEFPARM file as the IHFS SHEF decoder. This file is located on the `rax` in the `/rfc_arc/cfg/decoders` directory. Starting with build `ob7.2`, the RFC can choose to point to the directory where the IHFS `shefdecoder` SHEFPARM file resides (the NFS mount point has been changed so that that location can now be seen by the `RAX`).

## 2.3 “Housecleaning” Requirements

Ensure that the `purge_files` script is housecleaning the directories defined by the `apps_defaults` tokens `adb_shef_raw_err_dir` and `adb_shef_raw_logs_dir`.

## 3.0 User How-To

### 3.1 Start and Stop Scripts

Start and stop scripts have been provided to the user. These scripts use a similar concept as the start and stop scripts for the IHFS SHEF decoder. These scripts can be found in the directory `/rfc_arc/scripts/decoders` and are called:

```
start_raw_decoder
stop_raw_decoder
```

These scripts can be run through `arcmenu --> Shefdecoders menu`, or on the command line. Any user of the `fxalpha` group can run them.

### 3.2 Parsing Errors/Warnings

The parsing portion of the `shef_decode_raw` program now uses the same parser as the IHFS SHEF decoder. The possible parsing warnings/errors are as follows:

1. not used
2. Two digits are required in date or time group

3. An expected parameter code is missing
4. File read error while accessing data file
5. No dot in column 1 when looking for new message
6. Dot found but not in column 1 of new message
7. Unknown message type, looking for .A, .B, or .E
8. Bad char in message type format (or missing blank delimiter)
9. Last message format was different from this continuation messg
10. Last message was NOT a revision unlike this continuation messg
11. Last message had an error so cannot continue
12. No positional data or no blank before it
13. Bad character in station id
14. Station id has more than 8 characters
15. Bad number in positional data date group
16. Incorrect number in date group
17. Incorrect number in time group
18. Missing blank char in positional data
19. Bad creation date
20. Bad date code letter after the character "D"
21. Unknown data qualifier (need E,F,R,Q,T,S,V or other additions)
22. Unknown data units code (need S or E)
23. Unknown duration code (need Y,M,D,H,N,S,Z and others)
24. Bad 2-digit number following duration code
25. Unknown time interval code (need Y,M,D,H,N,S,E)
26. Bad 2-digit number following time interval code
27. Bad character after "DR" (relative date code)
28. Bad 1- or 2-digit number in relative date code
29. Bad character in parameter code
30. Bad parameter code calls for send code
31. Trace for code other than PP, PC, PY, SD, SF, SW
32. Variable duration not defined
33. Bad character where delimiter is expected
34. Non-existent value for given type and source parameter code
35. ZULU, DR, or DI has send code QY, PY, or HY
36. Forecast data given without creation date
37. No value given after parameter code and before slash or eol
38. Explicit date for codes DRE or DIE is not the end-of-month
39. Year not in good range (1753-2199)
40. Exceeded limit of data items
41. Too many data items for given .B format
42. Not enough data items for given .B format
43. Cannot adjust forecast date to Zulu time
44. Time between 0201 & 0259 on day changing from std to daylight
45. No time increment specified (use DI code)
46. No ".END" message for previous ".B" format
47. ID requires 3 to 8 characters
48. For dayl savgs time, check Apr or Oct for 1976 thru 2040 only
49. Bad character in the message
50. Missing parameter code
51. Bad value chars (or missing delimiter), data may be lost
52. Bad chars in data value field
53. "?" not accepted, use "M" (or change program)
54. Parameter code is too long or too short
55. Missing delimiter between data type fields
56. Missing delimiter after data type field
57. Should use "/" instead of blank for delimiter
58. Parm codes PP and PC require decimal value

59. Abort, cannot read "shefparm" file correctly
60. Non-existent value for given duration parameter code
61. Non-existent value for given extremum parameter code
62. Non-existent value for given conversion factor parameter code
63. Non-existent value for given probability parameter code
64. Parameter code too short or field misinterpreted as param-code
65. Comma not allowed in data field, data value is lost
66. Date check for yr-mo-da shows bad date
67. No data on line identified with a message type format
68. An unexpected ".END" message was encountered
69. BUMMER!!! Maximum number of errors reached, abort message
70. Cannot output to binary shefpars file
71. Cannot access "PE conversion factors" from the "shefparm" file
72. Cannot access "send codes" from the "shefparm" file
73. Cannot access "duration codes" from the "shefparm" file
74. Cannot access "type/source codes" from the "shefparm" file
75. Cannot access "extremum codes" from the "shefparm" file
76. Cannot access "probability codes" from the "shefparm" file
77. Cannot read "SHEFPARM" file!!!!
78. Bad character in data value, value is lost
79. Julian day should be written with 3 digits
80. Too many digits in date group!
81. Too many characters in quotes
82. Data line found before completing .B format line(s)
83. Missing slash delimiter or bad time zone code
84. Too many chars in qualifier code, data value is lost
85. Bad data qualifier, rest of format is lost
86. Retained comment found without a data value, comment is lost
87. Unexpected slash found after parameter code, before data value
88. Cannot access "qualifier codes" from the "shefparm" file
89. not used
90. Unknown error number given

### 3.3 Posting Warnings/Errors

The Warning/Error messages depend on how some of the apps\_defaults tokens are set. Examples of posting error messages that are now available in this release are:

```
LDP1 is not defined in location table; data not posted.
BLDP1 is not defined in location table; data posted to unkstnvalue.
Ignoring duplicate value for HDDN4 - HGIRGZZ for obstime (validtime) 2005-03-28 15:15:00.
SVEV2 - HGIRGZZ not defined in ingestfilter table; data not posted.
SKVP1 - PCIRGZZ not defined in ingestfilter table; data posted to unkstnvalue.
KCDM7 - HGIRZZZ obstime '2005-02-28' > 10 days in the past; data not posted.
KCDM7 - HGIRZZZ obstime '2005-04-30 12:00' > 60 minutes in the future; data not posted.
```

Duplicate, location, ingestfilter and outside time window messages are sent to the individual message files placed in the directory /rfc\_arc/log/deccoder/raw/err.

Error messages related to Postgres may appear in the raw.out file. This file is located in the directory /rfc\_arc/logs/decoder/raw/logs. Normally the raw.out (and raw.err) files should be empty. Possible messages include not being able to insert a row into a table and not being able to update a row in a table.

## 4.0 Daily Log File

The shef\_decoder\_raw program generates a daily log file. The format of this log file is based on the IHFS SHEF decoder's daily log file as much as possible. An excerpt of a daily log file follows.

```

Processing file: RR3CYS.0328.000107; at 2005-03-28 00:01:21
  Header productId: KCYSRR3CYS; timeZ= 2005-03-28 00:00
  LogFile: KCYSRR3CYS.0328.000021
Parsing data.
  Parse errs/warn=  0 /  0
Posting data.
  6 records processed
Total PEDRSEP:      4 ValIns:    4 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEIRSEP:      0 ValuIns:   0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PECRSEP:      2 ValIns:    2 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEDFSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEDCSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PAIREDV:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEOOSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEHFSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEMRSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEQFSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total UNKSTNV:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total Outside Window:  0
Net Total:          6
PEDRSEP Unk Location:  0 Unk Ingestfilter:  0
PEIRSEP Unk Location:  0 Unk Ingestfilter:  0
PECRSEP Unk Location:  0 Unk Ingestfilter:  0
PEDFSEP Unk Location:  0 Unk Ingestfilter:  0
PEDCSEP Unk Location:  0 Unk Ingestfilter:  0
PAIREDV Unk Location:  0 Unk Ingestfilter:  0
PEOOSEP Unk Location:  0 Unk Ingestfilter:  0
PEMRSEP Unk Location:  0 Unk Ingestfilter:  0
PEHFSEP Unk Location:  0 Unk Ingestfilter:  0
PEQFSEP Unk Location:  0 Unk Ingestfilter:  0
End time (elapsed parse,post): 2005-03-28 00:01:22 (00:00, 00:01)
-----
Processing file: RR7RIW.0328.000139; at 2005-03-28 00:01:52
  Header productId: KRIWRR7RIW; timeZ= 2005-03-28 00:01
  LogFile: KRIWRR7RIW.0328.000152
Parsing data.
  Parse errs/warn=  0 /  0
Posting data.
  1 records processed
Total PEDRSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PECRSEP:      1 ValIns:    1 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEIRSEP:      0 ValuIns:   0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEDFSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEDCSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PAIREDV:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0
Total PEOOSEP:      0 ValIns:    0 ValUpd:   0 NoInsRec:  0 NoUpdRec:  0

```

F. SHEF Decoders

```
Total PEHFSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEMRSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEQFSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total UNKSTNV:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
```

Total Outside Window: 0

Net Total: 1

```
PEDRSEP Unk Location:      0 Unk Ingestfilter:      0
PEIRSEP Unk Location:      0 Unk Ingestfilter:      0
PECRSEP Unk Location:      0 Unk Ingestfilter:      0
PEDFSEP Unk Location:      0 Unk Ingestfilter:      0
PEDCSEP Unk Location:      0 Unk Ingestfilter:      0
PAIREDV Unk Location:      0 Unk Ingestfilter:      0
PEOOSEP Unk Location:      0 Unk Ingestfilter:      0
PEMRSEP Unk Location:      0 Unk Ingestfilter:      0
PEHFSEP Unk Location:      0 Unk Ingestfilter:      0
PEQFSEP Unk Location:      0 Unk Ingestfilter:      0
```

End time (elapsed parse,post): 2005-03-28 00:01:52 (00:00, 00:00)

-----  
Processing file: RR1MSO.0328.000138; at 2005-03-28 00:01:53

Header productId: KMSORR1MSO; timeZ= 2005-03-28 00:01

LogFile: KMSORR1MSO.0328.000153

Parsing data.

Parse errs/warn= 0 / 0

Posting data.

10 records processed

```
Total PEDRSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEIRSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PECRSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEDFSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEDCSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PAIREDV:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEOOSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEHFSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEMRSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total PEQFSEP:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
Total UNKSTNV:      0 ValIns:      0 ValUpd:      0 NoInsRec:      0 NoUpdRec:      0
```

Total Outside Window: 0

Net Total: 0

```
PEDRSEP Unk Location:      0 Unk Ingestfilter:      0
PEIRSEP Unk Location:      0 Unk Ingestfilter:      0
PECRSEP Unk Location:      0 Unk Ingestfilter:      0
PEDFSEP Unk Location:      0 Unk Ingestfilter:      0
PEDCSEP Unk Location:      0 Unk Ingestfilter:      0
PAIREDV Unk Location:      0 Unk Ingestfilter:      0
PEOOSEP Unk Location:      0 Unk Ingestfilter:      0
PEMRSEP Unk Location:      0 Unk Ingestfilter:      0
PEHFSEP Unk Location:      0 Unk Ingestfilter:      0
PEQFSEP Unk Location:      0 Unk Ingestfilter:      0
```

End time (elapsed parse,post): 2005-03-28 00:01:53 (00:00, 00:00)

-----

## 5.0 Message Error Files

The shef\_decoder\_raw program generates message error files. The format of these files is based on the IHFS SHEF decoder's message error files as much as possible. An example of this log file is shown below.

```

SRUS55 KRIW 311922
RR3RIW
.A CAPW4 0331 Z DH1921/SD 2
:sbn/krf

NUMBER OF WARNINGS .... 0
NUMBER OF ERRORS ..... 0

TOTAL NUMBER OF LINES .. 5
(parsing routines: ob4-r25)
CAPW4 - SDIRZZZ not defined in ingestfilter table; data posted to unkstnvalue.
1 records processed
Total PEDRSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PECRSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PEDFSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PEDCSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PAIREDV: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PEOOSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PEHFSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PEMRSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total PEQFSEP: 0 ValIns: 0 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total UNKSTNV: 1 ValIns: 1 ValUpd: 0 NoInsRec: 0 NoUpdRec: 0
Total Outside Window: 0
Net Total: 1
PEDRSEP Unk Location: 0 Unk Ingestfilter: 0
PECRSEP Unk Location: 0 Unk Ingestfilter: 1
PEDFSEP Unk Location: 0 Unk Ingestfilter: 0
PEDCSEP Unk Location: 0 Unk Ingestfilter: 0
PAIREDV Unk Location: 0 Unk Ingestfilter: 0
PEOOSEP Unk Location: 0 Unk Ingestfilter: 0
PEMRSEP Unk Location: 0 Unk Ingestfilter: 0
PEHFSEP Unk Location: 0 Unk Ingestfilter: 0
PEQFSEP Unk Location: 0 Unk Ingestfilter: 0
    
```

## 6.0 Posting Summary Information

The posting summary information that appears in both the daily log file and the message error files can be broken into three parts, these are: part 1 - status of posting to each of the various "raw" data value tables, part 2 - misc. totals, and part 3 - general summary information indicating why values were not posted.

	←	↑	→	↓	○			
Part 1	Total PEDRSEP:	0	ValIns: 0	ValUpd: 0	NoInsRec: 0	NoUpdRec: 0		
	Total PECRSEP:	0	ValIns: 0	ValUpd: 0	NoInsRec: 0	NoUpdRec: 0		
	Total PEDFSEP:	0	ValIns: 0	ValUpd: 0	NoInsRec: 0	NoUpdRec: 0		
	Total PEDCSEP:	0	ValIns: 0	ValUpd: 0	NoInsRec: 0	NoUpdRec: 0		
	Total PAIREDV:	0	ValIns: 0	ValUpd: 0	NoInsRec: 0	NoUpdRec: 0		
	Total PEOOSEP:	0	ValIns: 0	ValUpd: 0	NoInsRec: 0	NoUpdRec: 0		

```

Total PEHFSEP:    0 ValIns:    0 ValUpd:    0 NoInsRec:    0 NoUpdRec:    0
Total PEMRSEP:    0 ValIns:    0 ValUpd:    0 NoInsRec:    0 NoUpdRec:    0
Total PEQFSEP:    0 ValIns:    0 ValUpd:    0 NoInsRec:    0 NoUpdRec:    0
Total UNKSTNV:    1 ValIns:    1 ValUpd:    0 NoInsRec:    0 NoUpdRec:    0
    
```

Part 2

```

Total Outside Window:    0
Net Total:                65
    
```

Part 3

```

←                               ↑
PEDRSEP Unk Location:    0 Unk Ingestfilter:    0
PECRSEP Unk Location:    0 Unk Ingestfilter:    1
PEDFSEP Unk Location:    0 Unk Ingestfilter:    0
PEDCSEP Unk Location:    0 Unk Ingestfilter:    0
PAIREDV Unk Location:    0 Unk Ingestfilter:    0
PEOOSEP Unk Location:    0 Unk Ingestfilter:    0
PEMRSEP Unk Location:    0 Unk Ingestfilter:    0
PEHFSEP Unk Location:    0 Unk Ingestfilter:    0
PEQFSEP Unk Location:    0 Unk Ingestfilter:    0
    
```

### **Part 1**

This part currently consists of ten lines, one line for each of the tables the shef\_decode\_raw program currently can post to. Each row consists of five values.

Column 1 indicates the total number of values inserted and/or updated to this table.

Column 2 indicates the total number of values inserted.

Column 3 indicates the total number of values updated.

Column 4 indicates the total number of records where insert was attempted but failed.

Column 5 indicates the total number of records where update was attempted but failed.

Columns 4 and 5 should always have zero totals; otherwise there is a problem.

### **Part 2**

This part currently consists of two lines.

Row 1 indicates the number of values where the observation time was outside the window defined by the apps\_defaults tokens *adb\_shef\_winpast* & *adb\_shef\_winfuture*.

Row 2 indicates the total number of values posted for all the tables.

### **Part 3**

Similar to part 1, this part currently consists of nine lines, one line for each of the tables (except *unkstnvalue*) that the shef\_decode\_raw program can currently post to. It does not include information for the *unkstnvalue* table as the posting to that table is controlled by the apps\_defaults token *adb\_shef\_raw\_post\_unk*. Each row consists of two values.

Column 1 indicates the total number of values that could not be posted to that table because the lids were not in the location table.

Column 2 indicates the total number of values that could not be posted to that table because the lid and/or SHEF pedtse code was not in the *ingestfilter* table.

## 7.0 Troubleshooting Information

Check the raw.out, raw.err, daily log and message error files. If the user cannot determine the source of the problem by viewing these files, contact the RFC Support Group for assistance.

## 8.0 References

NWS Directive 10-942 Standard Hydrometeorological Exchange Format (SHEF) Manual

RFC Archive DB Team Request for Change to SHEF submitted February 28, 2002.

Website with information on SHEF and the IHFS shefdecoder application  
<http://www.nws.noaa.gov/os/whfs/shef.shtml>