F-2. shef_decode_pro

1.0 General Information

The shef_decode_pro program posts data to the “processed” data value tables. This program is run by oper as a background process and is constantly monitoring the /rfc_arc_data/q/processed directory for incoming messages. These incoming SHEF messages can come from various sources, however, the primary source are the level 1 processors run by oper’s cron. The application is written in Fortran and ecpg/C.

The application writes to the following SHEF data value tables: *pehpsep*, *pempsep*, *peqpsep*, and *pedpsep*.

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 session 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 ds (now dx ??) and be mounted to the rax, while the shef_decode_pro incoming directory will remain on the rax with no mount back to the dx (formerly ds).

1.2 Enhancements/Bug Fixes/Changes

<u>Build OB7.2</u>

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

Enhancements

- one new apps_defaults token, which is:

```
        adb_shef_pro_tmp_dir   :   $(adb_pro_que)
```

Build OB6

Documentation

Sections 1.2, 2.1 and 6.0 have been updated.

Enhancements

- Added ability to shef_decode_pro to check the *location* and *ingestfilter* tables before posting data. This option works exactly as it does in shef_decode_raw (including messaging and counters). The check of *location* and *ingestfilter* is performed only if the new token *adb_shef_pro_checktab* is set to 'ON'. NOTE: If both *adb_shef_raw_checktab* and *adb_shef_pro_checktab* are set to 'ON', the two decoders may slow down dramatically. It is recommended that the *adb_shef_pro_checktab* be set to 'OFF' unless you are certain that you need the checks to be performed.
- Implemented part 3 counters for shef_decode_pro as part of the enhancement listed above for the ability to check *location* and *ingestfilter* tables.
- Implemented the *unkstnvalue* table for the shef_decode_pro as part of the enhancement listed above for the ability to check *location* and *ingestfilter* tables.

Bug Fixes

- Added a check in the shef_decode_pro for whether the value to post has an obstime/validtime that is outside the window specified by the tokens *adb_shef_winpast_pro* and *adb_shef_winfuture_pro*. The checks against *adb_shef_winpast_pro* and *adb_shef_winfuture_pro* are performed independently, and if either is set to '9999' the corresponding obstime/validtime check is not performed. [HSD Bug r26-21]
- ER-9. Added two tokens, *adb_shef_pro_locmess* and *adb_shef_pro_elgmess* that allow for user controlled messaging in the shef_decode_pro.
- 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 show the current decoders appear to change any year that is before 1970 to 1970 when it posts the date. [HSD Bug r26-43]

Build OB5

Documentation

Sections 1.2, 2.1, 3.3, 4.0, 5.0 and 6.0 have been updated. Section 1.3 (Known Limitations/Bugs) has been eliminated.

Enhancements

- ER-9. Added limited user controlled posting error/warning message capability. There is one new apps_defaults token, *adb_shef_pro_dupmess*.
- ER-1, R1-19. Implemented *pempsep* table in shef_decode_pro. Monthly data (i.e. data with a duration of 'M' in the pedtsep) will be posted to this table.

- Added error messaging to log file whenever a query of the archive database fails unexpectedly.
- R1-24, R1-30. New apps_defaults token *adb_shef_duplicate_pro*
- R1-45. Updated the SHEFPARM file.

Bug Fixes

- R1-24, R1-30. The SHEF decoder now correctly enforces the SHEF revision flag.
- 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.
- R1-41. Fixed problem with insert/update counts in the summary information.
- R1-43. The decoder can now overwrite existing values in the database with missing values.
- R1-44. Fixed shef_decode_pro to post max/min temperature data to the correct tables. If the duration code is instantaneous, data is posted as follows:

  IF (extreumum is ‘X’ or ‘N’) THEN post to *prodly*
  IF (extreumum is ‘R’ or ‘H’) THEN post to *proqtrly*
  IF (extreumum is ‘V’ or ‘L’) THEN post to *promnthly*
  OTHERWISE post to *prohrly*
- 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.
- Unnumbered Bug. Made the SHEF 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.

<u>Build OB4</u>

Enhancements

See section 1.1 about the changes.

Bug Fixes

- r1-6. Both the shef_decode_raw and shef_decode_pro programs end unexpectedly.

Changes

- changes in apps_defaults tokens
- 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

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

```
adb_name       : adb_ob5rha    # archive database name
adb_server     : adbs          # archive server name
adb_dir        : /rfc_arc                     # Base RFC Archive Directory
adb_pro_que    : /rfc_arc_data/q/processed/   # pathname for processed 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_pro   : 9999           # number of days in past to post data
adb_shef_winfuture_pro : 9999           # 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_pro_logs_dir  : $(adb_logs_dir)/decoder/processed/logs # pathname for the
                                                                # daily logs directory
adb_shef_pro_err_dir   : $(adb_logs_dir)/decoder/processed/err  # pathname for the
                                                                # product logs
                                                                # directory
adb_shef_pro_keeperror : IF_ERROR          # keep files (=ALWAYS) or only
                                           # when errors occur (=IF_ERROR)
adb_shef_pro_dupmess   : ON                # ON/OFF
adb_shef_duplicate_pro : USE_REVCODE       #  ALWAYS_OVERWRITE/USE_REVCODE
adb_shef_pro_perflog   : OFF               # OFF/ON
```

```
adb_shef_pro_locmess : ON     # ON/OFF ON only works if adb_shef_pro_checktab
                              # is also ON
adb_shef_pro_elgmess : OFF    # ON/OFF
adb_shef_pro_checktab : OFF   # ON/OFF ON - check lid and ingestfilter
                              # before posting  OFF - post all valid data
                              # do not check location and ingestfilter tables
adb_shef_pro_post_unk : NONE  # IDS_AND_DATA ..posts to same table as
                              # raw shefdecoder
```

adb_shef_pro_tmp_dir   : $(adb_pro_que)

<u>The following tokens are new as of build OB5:</u>

adb_shef_pro_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 that what is in the database, but the pedtsep and time are the same). Messages will only be written if *adb_shef_duplicate_pro* 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_duplicate_pro : 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 processed 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 processed SHEF decoder always updates duplicate data.

```
adb_shef_pro_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 processed decoder.

<u>The following tokens are new in build OB6:</u>

```
adb_shef_pro_checktab : OFF    # ON/OFF ON - check lid and ingestfilter
                               # before posting  OFF - post all valid data
                               # do not check location and ingestfilter tables
```

NOTE: If both *adb_shef_raw_checktab* and *adb_shef_pro_checktab* are set to ‘ON’, the two decoders may slow down dramatically. It is recommended that the *adb_shef_pro_checktab* be set to ‘OFF’ unless you are certain that you need the checks to be performed.

```
adb_shef_pro_post_unk :   NONE     # IDS_AND_DATA/NONE – NONE = do not post
                                   # to UnkStnValue table, IDS_AND_DATA = post
                                   # to UnkStnValue table
```

```
adb_shef_pro_locmess :  OFF                # ON/OFF
```

If set to ‘ON’, then a message will be displayed whenever 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_pro_post_unk* is set to ‘IDS_AND_DATA’, or it will state that the value was discarded if *adb_shef_pro_post_unk* is set to ‘NONE’.

```
adb_shef_pro_elgmess :  OFF                # ON/OFF
```

If set to ‘ON’, then a message will be displayed whenever 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_pro_post_unk* is set to 'IDS_AND_DATA', or it will state that the value was discarded if *adb_shef_pro_post_unk* is set to 'NONE'.

<u>The following token is new as of build OB7.2:</u>

adb_shef_pro_tmp_dir : $(adb_pro_que) # 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. Although the problem never occurred with the shef_decode_pro application, the feature was added just in case a problem should occur in the future.

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_pro_err_dir* and *adb_shef_pro_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_processed_decoder
stop_processed_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_pro 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 stnd 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 Warning/Error Examples

The warning/error messages depend on how some of the apps_defaults tokens are set. An example of a posting error message that is now available in this release is:

```
Ignoring duplicate value for HDDN4 – HGI1GZZ for obstime (validtime) 2005-03-
28 15:15:00.
```

Duplicate messages are sent to the individual message files placed in the directory /rfc_arc/log/decoder/processed/err

Otherwise, posting errors are limited to providing the Postgres error information in the daily log file. Miscellaneous information may be found in the pro.out and/or pro.err log files. These files are located in the directory /rfc_arc/logs/decoder/processed/logs. Normally the pro.out and pro.err files should be empty.

4.0 Daily Log File

The shef_decode_pro 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 from a daily log file follows.

```
Processing file: h_trans.shef.2004; at 2005-03-30 15:44:25
    Header productId: KRFCHG2HG; timeZ= 2005-03-30 13:32
    LogFile: KRFCHG2HG.0330.133225
Parsing data.
    Parse errs/warn=   0 /   0
Posting data.
  35136 records processed
Total PEHPSEP:  35136  ValIns: 1464  ValUpd: 33672  NoInsRec:    0  NoUpdRec:    0
Total PEDPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEMPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEQPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total Outside Window:     0
Net Total:    35136
PEHPSEP Unk Location:     0   Unk Ingestfilter:     0
PEDPSEP Unk Location:     0   Unk Ingestfilter:     0
PEMPSEP Unk Location:     0   Unk Ingestfilter:     0
PEQPSEP Unk Location:     0   Unk Ingestfilter:     0
End time (elapsed parse,post): 2005-03-30 15:44:39 (00:00, 00:14)
----------
Processing file: h_trans.shef.2003; at 2005-03-30 15:44:39
    Header productId: KRFCHG2HG; timeZ= 2005-03-29 21:51
    LogFile: KRFCHG2HG.0329.215139
Parsing data.
    Parse errs/warn=   0 /   0
Posting data.
  35040 records processed
Total PEHPSEP:  35040  ValIns: 1460  ValUpd: 33580  NoInsRec:    0  NoUpdRec:    0
Total PEDPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEMPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEQPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total Outside Window:     0
Net Total:    35040
PEHPSEP Unk Location:     0   Unk Ingestfilter:     0
PEDPSEP Unk Location:     0   Unk Ingestfilter:     0
PEMPSEP Unk Location:     0   Unk Ingestfilter:     0
PEQPSEP Unk Location:     0   Unk Ingestfilter:     0
End time (elapsed parse,post): 2005-03-30 15:44:51 (00:01, 00:11)
----------
Processing file: h_trans.shef.2002; at 2005-03-30 15:44:51
    Header productId: KRFCHG2HG; timeZ= 2005-03-29 21:44
    LogFile: KRFCHG2HG.0329.214451
Parsing data.
    Parse errs/warn=   0 /   0
Posting data.
  35040 records processed
Total PEHPSEP:  35040  ValIns: 1460  ValUpd: 33580  NoInsRec:    0  NoUpdRec:    0
Total PEDPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEMPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEQPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total Outside Window:     0
Net Total:    35040
PEHPSEP Unk Location:     0   Unk Ingestfilter:     0
PEDPSEP Unk Location:     0   Unk Ingestfilter:     0
PEMPSEP Unk Location:     0   Unk Ingestfilter:     0
PEQPSEP Unk Location:     0   Unk Ingestfilter:     0
End time (elapsed parse,post): 2005-03-30 15:45:02 (00:00, 00:11)
```

5.0 Message Error Files

The shef_decode_pro 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 excerpt from a daily log file is shown below.

```
SRUS55 KRFC 291958
 HG2HG
 .E CHZM7    19940101 DH00/HGI1GZZ/DIH1/-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E1 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E2 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E3 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000
 .E CHZM7    19940102 DH00/HGI1GZZ/DIH1/-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E1 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E2 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E3 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000
 .E CHZM7    19940103 DH00/HGI1GZZ/DIH1/-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E1 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E2 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E3 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000
 .E CHZM7    19940104 DH00/HGI1GZZ/DIH1/-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E1 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E2 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E3 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000
 .E CHZM7    19940105 DH00/HGI1GZZ/DIH1/-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E1 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E2 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /
 .E3 -9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000 /-9999.000
…
```

etc.

```
….
.E SNZM7    19941231 DH00/HGI1GZZ/DIH1/   8.800S/-9999.000 /   8.790S/-9999.000 /
 .E1    8.780S/-9999.000 /   8.780S/-9999.000 /   8.770S/-9999.000 /-9999.000 /
 .E2 -9999.000 /-9999.000 /-9999.000 /   8.750S/-9999.000 /   8.750S/-9999.000 /
 .E3    8.730S/-9999.000 /   8.730S/-9999.000 /   8.730S/-9999.000

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

    TOTAL NUMBER OF LINES ..    2923
    (parsing routines: ob4-r25)
  17520 records processed
Total PEHPSEP:  17520  ValIns:    0  ValUpd: 17520  NoInsRec:    0  NoUpdRec:    0
Total PEDPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEMPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEQPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total Outside Window:     0
Net Total:    17520
PEHPSEP Unk Location:     0   Unk Ingestfilter:     0
PEDPSEP Unk Location:     0   Unk Ingestfilter:     0
PEMPSEP Unk Location:     0   Unk Ingestfilter:     0
PEQPSEP 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 messages error files can be broken into three parts: part 1) status of posting to each of the various “processed” data value tables, part 2) miscellaneous totals, and part 3) general summary information indicating why a value was not posted.

Part 1

```
         ←                   ↑           →           ↓               ○
Total PEHPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEDPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEMPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total PEQPSEP:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
Total UNkSTNV:      0  ValIns:    0  ValUpd:    0  NoInsRec:    0  NoUpdRec:    0
```

Part 2

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

Part 3

```
                 ←                            ↑
PEHPSEP Unk Location:     0   Unk Ingestfilter:     0
PEDPSEP Unk Location:     0   Unk Ingestfilter:     0
PEMPSEP Unk Location:     0   Unk Ingestfilter:     0
PEQPSEP Unk Location:     0   Unk Ingestfilter:     0
```

<u>Part 1</u>

This part currently consists of five lines, one line for each of the tables the shef_decode_pro program currently can post to. Each line consists of 5 values.

Column 1 indicates the total number of values inserted and/or updated in 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.

<u>Part 2</u>

This part currently consists of two lines.

Line 1 indicates the number of records where the observation time was outside the window defined by the apps_defaults tokens *adb_shef_winpast_pro* and *adb_shef_winfuture_pro.*
Line 2 indicates the total number of values posted for all the tables.

Part 3

This part is used as of build OB6 if the apps_defaults token *adb_shef_pro_checktab* is set to ON.

Similar to part 1, this part currently consists of four lines, one line for each of the tables (except *unkstnvalue*) that the shef_decode_pro program currently can 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_pro_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 pro.out, pro.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 Manual 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