

```

1:
2:  /*****
3:  NAME:      ATL_utilities
4:  PURPOSE:   This package contains computational utilites necessary to load and
5:             maintain the Norpac data set.  Previous iterations of Norpac have
6:             embedded these functions within GUI Modules and or triggers.  As the
7:             redevelopment of the Observer software continues.  Those processes
8:             are to be migrated here and called from the various modules which
9:             require them.  Excepted, are those validation functions or processes
10:            which are unique to the module, table, or view.
11:
12:  Naming Conventions;
13:     Parameters begin p_
14:     Local Variables  v_
15:     Global Variables g_
16:
17:  REVISIONS:
18:  Ver      Date      Author      Description
19:  -----
20:  1.0      01/22/2007  Doug Turnbull  1. Created this package.
21:  *****/
22:
23:  /*****
24:  Depth may be reported in Meters or fathoms the following paired functions do the
25:  conversion and return a whole number (Rounded) value.
26:  *****/
27:  FUNCTION Convert_Meters_To_Fathoms(p_Meters IN NUMBER) RETURN NUMBER;
28:  FUNCTION Convert_Fathoms_To_Meters(p_Fathoms IN NUMBER) RETURN NUMBER;
29:
30:  /*****
31:  Latitude and Logitude have historically been reported to the precision of minutes.
32:  Atlas now optionally permits seconds to be reported.  If seconds are known then that
33:  precision in the conversion to decimal degrees shall be maintained.  If a NULL is
34:  passed to the function then decimal degrees are returned with minutes precision.
35:  The calling program must convert longitude reported to the west of the 180th meridian
36:  as negative.
37:  *****/
38:  FUNCTION Convert_Long_DMS_To_DD (p_degrees IN NUMBER, p_minutes IN NUMBER,
39:  p_seconds IN NUMBER, p_ew IN VARCHAR2) RETURN NUMBER;
40:
41:  FUNCTION Convert_Lat_DMS_To_DD (p_degrees IN NUMBER, p_minutes IN NUMBER, p_seconds IN NUMBER)
42:  RETURN NUMBER;
43:
44:  /*****
45:  The following group of functions use the Oracle Spatial Cartridge to map latitude and
46:  logitude positions to NMFS Reporting and Special areas.  The code has been derived
47:  from the sdo_columns_edit BIUR trigger on the edit haul table from the legacy system,
48:  and from the position package.  Both Created by Alison Vijgen.
49:  *****/
50:
51:  /*****

```

```

52:  The conversion of longitude is required of values to the east of the 180th meridian.
53:  Observer reporting identifies these as 179 degrees East for example. The configura-
54:  tion within Oracle Spatial uses a 360 degree globe - 179 E must therefore be submitted
55:  as 181. The calling program must, as in the conversion to decimal degrees submit latitude
56:  to the east of the 180th meridian as a negative number.
57:  *****/
58:  FUNCTION Longitude_Converted(p_degrees IN VARCHAR2, p_minutes IN VARCHAR2, p_seconds IN VARCHAR2,
59:  p_ew IN VARCHAR2)
60:  RETURN NUMBER;
61:
62:  *****
63:  The submission of Observer data sets to AKR requires the computation of Federal Reporting
64:  Area and Boolean identification of COBLZ. These data are not required for debriefing of
65:  observers and is more appropriately computed in Juneau. For the interim year or years
66:  of transition to a new process it will be continued to be computed and submitted with haul
67:  data. In both functions the calling program must, submit longitude to the west of the
68:  180th meridian as a negative number and it should be noted only the retrieval coordinates
69:  are used.
70:  *****/
71:
72:  FUNCTION Is_Coblz(p_latitude_degree IN NUMBER, p_latitude_minutes IN NUMBER,
73:  p_ew IN VARCHAR2, p_longitude_degree IN NUMBER, p_longitude_minutes IN NUMBER)
74:  RETURN BOOLEAN;
75:
76:  FUNCTION Is_Coblz_YN(p_latitude_degree IN NUMBER, p_latitude_minutes IN NUMBER,
77:  p_ew IN VARCHAR2, p_longitude_degree IN NUMBER, p_longitude_minutes IN NUMBER)
78:  RETURN VARCHAR2;
79:
80:  FUNCTION Get_RepAREA(p_latitude_degree IN NUMBER, p_latitude_minutes IN NUMBER,
81:  p_ew IN VARCHAR2, p_longitude_degree IN NUMBER, p_longitude_minutes IN NUMBER)
82:  RETURN NUMBER;
83:
84:  FUNCTION Get_RetrievalSHAPE(p_latitude_degree IN NUMBER, p_latitude_minutes IN NUMBER,
85:  p_ew IN VARCHAR2, p_longitude_degree IN NUMBER, p_longitude_minutes IN NUMBER)
86:  RETURN MDSYS.SDO_GEOMETRY;
87:
88:  *****
89:  When transforming ATL data for InSeason and Domestic vessel and plants uniquely identified
90:  Oracle sequences must be converted to Observer Codes. The following functions return the
91:  Code when passed the sequence number;
92:  *****/
93:  FUNCTION Get_VesselCode(p_vessel_seq IN NUMBER)
94:  RETURN VARCHAR2;
95:  -- For the ATL_species_composition View. The observer vessel code must be looked up
96:  -- when supplied a haul sequence number this
97:
98:  FUNCTION Get_HaulVesselCode(p_cruise IN NUMBER, p_haul_seq IN NUMBER)
99:  RETURN VARCHAR2;
100:
101:
102:  FUNCTION Get_Vessel_permit(p_vessel_seq IN NUMBER)

```

```

103:         RETURN VARCHAR2;
104:
105:     FUNCTION GET_PlantCode(p_plant_seq  IN NUMBER)
106:         RETURN VARCHAR2;
107:
108: /***** Get_FishingStartDate *****/
109:     Fishing start date is defined as the retrieval date of the first haul on a trip. This function
110:     is called by the Current_haul view.
111: *****/
112:     FUNCTION Get_FishingStartDate (p_cruise IN NUMBER, p_trip_seq  IN NUMBER)
113:         RETURN DATE;
114:
115: /***** GetHaulDateOfEntry *****/
116:     The current haul function requires the load date be presented to AKR. This function Retrieves
117:     That date from the ATL_Record_Status table when passed the cruise and haul sequence
118: *****/
119:     FUNCTION Get_HaulDateOfEntry (p_cruise IN NUMBER, p_haul_seq  IN NUMBER)
120:         RETURN DATE;
121:
122: /***** Log Load Errors *****/
123:     Data from field deployments of Atlas are initially loaded into unconstrained working
124:     tables. Those data are then loaded into the Atl_Production tables by a call to
125:     ATL_LOAD_NORPAC.load_atl_tables which reads the load_queue and loads the tables in the correct
126:     order as determined by FK dependencies. Since with few exceptions loading ought not be
127:     interrupted because of other than relational errors errors are logged and later resolved or
128:     overridden as determined by management. This procedure provides the logging service at the
129:     initial load to ATL production tables, ETL to Inseason, and ETL to Domestic.
130: *****/
131:     PROCEDURE Log_Load_Error(p_error_number IN NUMBER, p_cruise  IN NUMBER,
132:                             p_table_name  IN VARCHAR2, p_pk1_column_name  IN VARCHAR2,
133:                             p_pk1_column_value  IN VARCHAR2, p_pk2_column_name  IN VARCHAR2,
134:                             p_pk2_column_value  IN VARCHAR2, p_load_level  IN VARCHAR2,
135:                             p_comments  IN VARCHAR2);
136:
137: /*****
138:     NAME:         get_cruise
139:     PURPOSE:     Triggers populate the Record_Set_Status and Record_Status tables.
140:                 Since on the AFSC end sequence generated PK values submitted by
141:                 the deployed version of ATLAS may not be unique outside the context
142:                 of an observer cruise, the cruise must also be supplied with the
143:                 dataset. This function determines and return the cruise for
144:                 transaction tables that are not part of an arc.
145:
146:     REVISIONS:
147:     Ver         Date         Author         Description
148:     -----
149:     1.0         02/20/2007    Doug Turnbull    1. Initial Creation
150:                 03/03/2007    Doug Turnbull    2. Completed debugging
151:                 03/03/2007    Doug Turnbull    3. Completed initial unit testing
152:                 05/21/2007    Doug turnbull    4. Debugged Fish_Inv_Specimen length pathway
153:

```

```

154:      NOTES: This function when Passed the Current Table name, and the Parent primary key values
155:      by the calling trigger, walks up the FK thread to identify the observer cruise which
156:      is the return value. For the set of transaction tables foreign keys cascade. Since this
157:      precludes quering the parent table(s) which are in an indeterminate state a mutating table
158:      error ORA-04091 must be trapped. To allow handling this error the PK value of the current
159:      table is also passed to this function P_VALUE3. The record status can then be directly set
160:      to D (delete). If no data is found when trapping the error, nothing is recorded as the
161:      on delete cascade FK function will flow through the system on the NOPAC data set as well
162:      obviating the need to log this in the record status. The function returns NULL in this case.
163:      Alternatively this feature could be exploited to simplify the passed values to AFSC in the
164:      record status table, deleting from the passed set of data those records affected by the
165:      cascade delete FK. Something to think about for the next release.
166:
167:      *****/
168: FUNCTION GET_CRUISE (P_TABLE IN VARCHAR2, P_VALUE1 IN NUMBER, P_VALUE2 IN NUMBER, P_VALUE3 IN NUMBER )
169:      RETURN ATL_OBSERVER_CRUISE.CRUISE%TYPE;
170:
171:
172: /***** Extrapolate_Fixed_Species *****/
173:      This procedure when passed a cruise, haul, species combination returns the extrapolated weight
174:      and count for a longline or pot operation for that species, for that haul. Note that the values
175:      are accumulated for a species and sample by haul.
176:
177:      Rules:
178:          Required for a Sample:
179:              combined_flag default = 'N'
180:              presorted_flag default = 'N'
181:              sample_hooks_pots required on parent samples, ignored on sub samples
182:          Required for a Species Comp: species_number required on parent samples, ignored on sub samples,
183:          Parent samples determine the total sampled hooks, and total numbers of a species.
184:          Subsamples determine the aggregated average weight of a species for a haul/trip/cruise
185:          The subsample will not have a total number of hooks. The species
186:          composition weights and numbers are required. The computed average weight of
187:          a species within the subsample is applied compute those missing values.
188:          If the species composition set is still missing values then
189:          the average weight for that species is aggregated from other hauls on the
190:          current trip.
191:          If the species composition set is still missing values then
192:          the average weight for that species is aggregated from other hauls on the
193:          current cruise.
194:          If the species composition set is still missing values then
195:          have a beer and think about it some more.
196:      *****/
197: PROCEDURE extrapol_fixed_gear_species(p_haul IN NUMBER, p_cruise IN NUMBER,
198:      p_extrapolated_weight OUT NUMBER, p_extrapolated_count OUT NUMBER);
199:
200: /*****
201:      NAME:          is_specimen_num_unique
202:      PURPOSE:       When passed The PK value from the parent length, the specimen type,
203:                    and the desired specimen number verify uniqueness of the specimen
204:                    number within a cruise/permit, species, and specimen type.

```

```

205:
206:     REVISIONS:
207:     Ver      Date          Author          Description
208:     -----  -
209:     1.0      8/16/2007      Doug Turnbull  1. Created this function.
210:           10/11/2007 Gary Zhou      5. Rewrote using a single nested query
211:     NOTES:
212:
213:     Automatically available Auto Replace Keywords:
214:     Object Name:    is_specimen_num_unique
215:     Sysdate:       8/16/2007
216:     Date and Time: 8/16/2007, 4:04:16 PM, and 8/16/2007 4:04:16 PM
217:
218:     *****/
219:
220: FUNCTION IS_SPECIMEN_NUM_UNIQUE
221: (P_LENGTH_SEQ IN atl_fish_inv_specimen.length_seq%TYPE
222: ,P_SPECIMEN_TYPE IN atl_fish_inv_specimen.specimen_type%TYPE
223: ,P_SPECIMEN_NUMBER IN atl_fish_inv_specimen.specimen_number%TYPE
224: ,P_SPECIES_CODE IN NUMBER
225: )
226: RETURN NUMBER;
227:
228: /*****/
229:     NAME:      CLEAR_PERCENT_RETAINED
230:     PURPOSE:   When passed The PK value from the parent sample and the species
231:               code, check to see if no others exist for this haul, and if not
232:               delete the percent retained record.
233:
234:     REVISIONS:
235:     Ver      Date          Author          Description
236:     -----  -
237:     1.0      10/25/2007      Doug Turnbull  1. Created this procedure.
238:
239:     NOTES:
240:     *****/
241: PROCEDURE CLEAR_PERCENT_RETAINED
242:
243: (P_SPECIES_CODE IN atl_species_composition.species_code%TYPE
244: ,P_SAMPLE_SEQ IN atl_sample.sample_seq%TYPE);
245:
246: /*****/
247:     NAME:      get_samples_haul_offload
248:     PURPOSE:   When passed The sample_seq get its haul/offload ancestor
249:
250:
251:     REVISIONS:
252:     Ver      Date          Author          Description
253:     -----  -
254:     1.0      12/11/2007      Ed Dunn        1. Created this procedure.
255:

```

```

256:      NOTES:
257:      *****/
258:  PROCEDURE GET_SAMPLES_HAUL_OFFLOAD
259:    (P_SAMPLE_SEQ IN atl_sample.sample_seq%TYPE
260:    ,P_CRUISE IN atl_sample.cruise%TYPE
261:    ,P_HAUL_SEQ OUT atl_haul.haul_seq%TYPE
262:    ,P_OFFLOAD_SEQ OUT atl_offload.offload_seq%TYPE);
263:
264:  /*****
265:  FUNCTION COMPUTE PROCESSOR CODE
266:    This utility is provided for the current_haul_mv materialized view.
267:    When passed a cruise and haul_seq, it looks up the vessel/plant permit,
268:    pads the value to five places and appends an initial value depending
269:    on vessel type:
270:      'P' for catcher processor (vessel type 1)
271:      'M' for mothership (vessel types 2 and 4)
272:      'F' for shoreside plants or floating processors
273:  *****/
274:  FUNCTION compute_processor_code
275:    (p_cruise IN atl_haul.cruise%TYPE,
276:    p_haul_seq IN atl_haul.haul_seq%TYPE)
277:    RETURN VARCHAR2;
278:
279:
280:  /*****
281:  FUNCTION GET PLANT PERMIT
282:    This utility supports the current_haul_mv materialized view.
283:    When passed a plant_seq, it returns the federal permit number
284:    of the processing plant associated with the plant sequence number.
285:  *****/
286:  FUNCTION get_plant_permit
287:    (p_plant_seq IN NUMBER)
288:    RETURN VARCHAR2;
289:
290:  /*****
291:  FUNCTION COMPUTE SAMPLE NUMBER
292:    This utility sums the sample numbers (counts) for a species for a haul.
293:    Its purpose is to support the current_spcomp_mv for the Region.
294:    For completeness, species apportioned back to a haul from an offload
295:    are included.
296:  *****/
297:  FUNCTION compute_sample_number
298:    (p_cruise IN atl_haul.cruise%TYPE,
299:    p_haul_seq IN atl_haul.haul_seq%TYPE,
300:    p_species_code IN atl_species_composition.species_code%TYPE,
301:    p_sex_code IN atl_species_composition.sex_code%TYPE)
302:
303:    RETURN NUMBER;
304:
305:  /*****
306:  FUNCTION COMPUTE TOTAL SAMPLE SIZE

```

```

307:      This utility sums all sample weights for a haul. Its purpose is to support
308:      the current_spcomp_mv for the Region.
309:      *****/
310:  FUNCTION compute_total_sample_size
311:      (p_cruise      IN atl_haul.cruise%TYPE,
312:       p_haul_seq    IN atl_haul.haul_seq%TYPE)
313:
314:      RETURN NUMBER;
315:
316:  /*****
317:  FUNCTION COMPUTE SAMPLE WEIGHT
318:      This utility sums the sample weights for a species for a haul.
319:      Its purpose is to support the current_spcomp_mv for the Region.
320:      For completeness, species apportioned back to a haul from an offload
321:      are included.
322:      *****/
323:  FUNCTION compute_sample_weight
324:      (p_cruise      IN atl_haul.cruise%TYPE,
325:       p_haul_seq    IN atl_haul.haul_seq%TYPE,
326:       p_species_code IN atl_species_composition.species_code%TYPE,
327:       p_sex_code     IN atl_species_composition.sex_code%TYPE)
328:
329:      RETURN NUMBER;
330:
331:  /*****
332:  FUNCTION COMPUTE TRAWL OTC
333:      This utility sums the sample weights for a sampled haul and returns this
334:      as a proxy for an official_total_catch.
335:      Its purpose is to support the current_spcomp_mv for the Region.
336:      For completeness, species apportioned back to a haul from an offload
337:      are included.
338:      *****/
339:  FUNCTION compute_trawl_otc
340:      (p_cruise      IN atl_extrawl_species_comp.cruise%TYPE,
341:       p_haul_seq    IN atl_extrawl_species_comp.haul_seq%TYPE)
342:
343:      RETURN NUMBER;
344:
345:  FUNCTION compute_sample_size
346:      (p_cruise      IN atl_extrawl_species_comp.cruise%TYPE,
347:       p_haul_seq    IN atl_extrawl_species_comp.haul_seq%TYPE)
348:
349:      RETURN NUMBER;
350:
351:  /*****
352:  FUNCTION get_cg_number
353:      This utility has two flavors to support the PACFIN CURRENT HAUL VIEW,
354:      When passed the domestic/inseason vessel from old norpac it looks up the
355:      CG number from vessplnt. When passes the vessel_seq from the ATL tables, it
356:      looks up the CG number from the ATL_lov_vessel table.
357:      *****/

```

```

358: FUNCTION get_cg_number
359: (p_table IN VARCHAR2,
360:  p_vessel IN VARCHAR2)
361:
362: RETURN VARCHAR2;
363:
364:
365: END ATL_utilities;
366: /
367:
368:
369: g_atlas_version NUMBER := 8.0;
370:
371: /*****
372: NAME: ATL_LOAD_NORPAC
373: PURPOSE: This Package Contains the ATL_ table load and Current View Presentation Processes
374: from the Atlas 2008 Application to the legacy Norpac data set. Note that some
375: of the detail from Observer Data collected and transmitted from deployed
376: versions of ATLAS is lost in the current view presentation layer.
377:
378: REVISIONS:
379: Ver Date Author Description
380: -----
381: 1.0 12/26/2006 Doug Turnbull 1. Created this package body as a development stub.
382: 3/20/2007 Doug Turnbull 2. Added Procedure Load_ATL_Tables
383: 3/25/2007 Doug Turnbull 3. Added Private Function verify_cruise_exists
384: 4/30/2007 Doug Turnbull 4. Completed Current_Haul
385: 6/8/2007 Doug Turnbull 5. Completed Current_SpComp
386: *****/
387: /*****
388: The extrapolation algorithms for species wieght and count are required to maintain the current
389: views for AKR fishery management. The values are presented in the view ATL_CURRENT_SPCOMP.
390: The following functions call the private procedures which compute and return the expanded
391: weights and numbers. These public functions provide the vehicle for display by the view.
392: *****/
393:
394: FUNCTION extrap_fixed_species_wt(p_cruise IN NUMBER, p_haul IN NUMBER,
395: p_species IN NUMBER, p_sex_code IN VARCHAR2)
396: RETURN NUMBER;
397:
398: FUNCTION extrap_fixed_species_ct(p_cruise IN NUMBER, p_haul IN NUMBER,
399: p_species IN NUMBER, p_sex_code IN VARCHAR2)
400: RETURN NUMBER;
401:
402: FUNCTION compute_fixed_OTC(p_cruise IN NUMBER, p_haul IN NUMBER)
403: RETURN NUMBER;
404:
405:
406: /*****
407: Samples are not of a specifc type in the Atl table structure. To support that notion
408: the structure of the sample/subsample must be examined and the results interpreted.

```



```

409:
410:     This function does the analysis and return the sample type code expected by Domestic
411:     or Inseason
412:     *****/
413:     FUNCTION compute_sample_type(p_cruise IN NUMBER, p_haul_seq IN NUMBER)
414:     RETURN VARCHAR2;
415:
416: *****/
417:     This function returns the haul_number when passed the haul_seq
418:     *****/
419:     FUNCTION get_haul_number(p_cruise IN NUMBER, p_haul_seq IN NUMBER)
420:     RETURN NUMBER;
421: *****/
422:     This function returns the number of distinct species in a sample (sample_number)
423:     *****/
424:     FUNCTION compute_number_of_species(p_cruise IN NUMBER, p_species_code IN VARCHAR2,
425:     p_sample_seq IN NUMBER, p_haul_seq IN NUMBER )
426:     RETURN NUMBER;
427:
428: *****/
429:     This function returns the percentage of a species retained for a haul
430:     *****/
431:     FUNCTION get_percent_retained(p_cruise IN NUMBER, p_haul_seq IN NUMBER,
432:     p_species_code IN NUMBER)
433:     RETURN NUMBER;
434:
435: *****/
436: PROCEDURE LOAD_ATL_TABLES
437:     This procedure creates a cursor for all records in the ATL Record Status table. This
438:     acts as a template to move records from the working tables into the Atlas production
439:     table set.
440:     Original Creation - March 15, 2007
441:     Data Error Trapping Added -
442:     *****/
443:     PROCEDURE load_atl_tables(p_return_code OUT NUMBER);
444:
445: *****/
446:     This function returns the load/edit date for a species composition record logged in the
447:     atl_record_status table. This is analogous to the date of entry for domestic and inseason
448:     and the value returned to the current view.
449:     *****/
450:     FUNCTION get_record_status_date(p_cruise IN NUMBER, p_table_name IN VARCHAR2,
451:     p_pk1_column_name IN VARCHAR2, p_pk1_value IN VARCHAR2,
452:     p_pk2_column_name IN VARCHAR2, p_pk2_value IN VARCHAR2)
453:     RETURN DATE;
454:
455: *****/
456:     This procedure updates the Record_Status when called by the AIUDR trigger that also creates
457:     the histories. It sets the status_code, editor, and modified date as appropriately depending
458:     on what action has been performed on the parent record. This provides the inseason advisor
459:     with last touch information - which could alternatively be derived from a query of the history

```

```

460:      table. The functionality is for the convenience of the user and is a recognized
461:      denormalization.
462:      *****/
463:      PROCEDURE set_record_edit_status(p_cruise IN NUMBER, p_table_name IN VARCHAR2,
464:          p_pk1_column_name IN VARCHAR2, p_pk1_value IN VARCHAR2,
465:          p_pk2_column_name IN VARCHAR2, p_pk2_value IN VARCHAR2,
466:          p_edit_date IN DATE, p_editor IN VARCHAR2, p_status_code IN VARCHAR2);
467:
468:      /*****
469:          This procedure updates the ATL_Record_Status of all members of a record set when called by
470:          the AIUDR trigger that also creates the histories for cruise vessel and cruise plant.
471:          It sets the debriefed flag when the debriefing process is complete (status = 3)
472:      *****/
473:      PROCEDURE set_record_edit_status(p_cruise IN NUMBER, p_table_name IN VARCHAR2,
474:          p_pk_value IN VARCHAR2);
475:
476:      /*****
477:          This procedure updates the ATL_Record_Status of all members of a record set when called by
478:          the AIUDR trigger on atlworking tables when data is entered from faxed data sheets
479:      *****/
480:      PROCEDURE set_record_edit_status(p_mode IN VARCHAR2,
481:          p_cruise IN NUMBER, p_table_name IN VARCHAR2,
482:          p_pk1_column_name IN VARCHAR2, p_pk1_value IN VARCHAR2,
483:          p_pk2_column_name IN VARCHAR2, p_pk2_value IN VARCHAR2,
484:          p_edit_date IN DATE, p_editor IN VARCHAR2, p_status_code IN VARCHAR2);
485:
486:      /*****
487:      NAME:          is_specimen_num_unique
488:      PURPOSE:      When passed The PK value from the parent length, the specimen type,
489:                   species and the desired specimen number verify uniqueness of the specimen
490:                   number within a cruise/permit, species, and specimen type.
491:      *****/
492:      FUNCTION is_specimen_num_unique(p_length_seq IN atl_fish_inv_specimen.length_seq%TYPE
493:          ,p_specimen_type IN atl_fish_inv_specimen.specimen_type%TYPE
494:          ,p_specimen_number IN atl_fish_inv_specimen.specimen_number%TYPE
495:          ,p_species_code IN NUMBER)
496:          RETURN BOOLEAN;
497:
498:
499:
500:
501:      END ATL_Load_norpac;
502:      /

```