

```

1 SET ECHO OFF;
2 SET FEEDBACK OFF;
3 SET VERIFY OFF;
4 SET TERMOUT ON;
5 SET SERVEROUT ON SIZE 500000;
6
7 -- !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
8 -- !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
9 -- *****
10 -- PROGRAM HISTORY *
11 -- *
12 -- WCUNEX_STATS_B.SQL *
13 -- IF A UNIT HAS NO EXPOSURE FOR ANY OF THE DESIGNATED CLASS CODES, NONE
  OF *
14 -- THE CLASS CODES NEED TO BE REPORTED. CODE 1111 IS THE ONLY CODE THAT *
15 -- GETS TRANSMITTED. THIS IS APPLICABLE TO ALL STATES EXCEPT NEW JERSEY. *
16 -- *
17 -- PROJECT.TASK DATE TECH. ACTION *
18 -- -----*
19 -- PLI32502.12477 30-SEP-2008 JPC INITIAL CREATION *
20 -- *
21 -- PLI32502.15785 08-SEP-2009 JPC REMOVE TABLE CREATE *
22 -- *
23 -- *
24 -- PLI32502.16827 04-JAN-2010 JPC Set troubleshooting displays *
25 -- *
26 -- PLI32502.16866 06-JAN-2010 JPC Add sqlcode for error-trapping *
27 --
28 -- PLI32502.18116 14-JUL-2010 JPC add logic for 9037 (DEVPX) in MA
29 --
30 -- PLI32502.19062 09-aug-2010 JPC Correct mod for 9037
31 -- *
32 -- PLI32502.19870 05-JAN-2011 JPC Add exposure transmit indicator*
33 --
34 -- PLI32502.21102 23-MAY-2011 JPC Add sequence; remove mod value
35 --
36 -- PLI32502.19866 25-JUL-2011 JPC Correct calculation for 9037
37 --
38 -- PLI32502.24578 14-JAN-2013 JPC Further corrections for 9037
39 --
40 -- PLI32502.26696 14-FEB-2014 JPC Switch Anniv date for poleff date
41 -- PLI32502.25454 11-APR-2014 JPC DevPx corrections
42 -- PLI32502.27967 01-AUG-2014 JPC added an NVL to the deviation percentage
43 -- PLI32502.28448 15-OCT-2014 JPC Fix DevPx for Mass.
44 -- *****
45 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
46
47 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
48 DECLARE
49 WS_SECTION VARCHAR2
50 (
51 12
52 )
53 ;
54 WS_UNPL_ID UNEX.UNEX_UNPL_WC_NO%TYPE;
55 WS_POLICY TUNX.POLICY_NUMBER%TYPE;

```

```

56 V_POLICY TUNX.POLICY_NUMBER%TYPE;
57 WS_CLASS UNEX.UNEX_WC_EXIT_CLASS_CODE%TYPE;
58 WS_HOLD_CLASS UNEX.UNEX_WC_EXIT_CLASS_CODE%TYPE;
59 WS_NETPX UNEX.UNEX_WC_EXIT_NET_PREM%TYPE;
60 WS_STDPX TUNX.STANDARD_PREMIUM%TYPE;
61 WS_W RTPX TUNX.WRITTEN_PREMIUM%TYPE;
62 WS_STAND TUNX.STANDARD_PREMIUM%TYPE;
63 V_DEVPX TUNX.WRITTEN_PREMIUM%TYPE;
64 V_RATE number;
65 V_DEV number;
66 V_MULTIPX number;
67 V_NOMODSUM TUNX.WRITTEN_PREMIUM%TYPE;
68 WS_MODPX UNEX.UNEX_WC_EXIT_MOD_NET_PREM%TYPE;
69 WS_PAYROLL TUNX.PAYROLL_AMOUNT%TYPE;
70 WS_STATE_NO UNEX.UNEX_STAT_GB_STATE_NO%TYPE;
71 WS_UNEX_WC_NO UNEX.UNEX_WC_NO%TYPE;
72 WS_SPLITNO UNEX.UNEX_WC_SPLIT_PERIOD_IND%TYPE;
73 WS_AFTERMOT UNEX.UNEX_WC_RECORD_TYPE%TYPE;
74 WS_MODAL UNEX.UNEX_WC_EXP_MODAL%TYPE;
75 WS_MODEFF UNEX.UNEX_WC_EXP_MODAL_EFF_DATE%TYPE;
76 WS_BURT UNEX.UNEX_WC_BURT_RATE%TYPE;
77 WS_BURT_EFF UNEX.UNEX_WC_BURT_EFF_DATE%TYPE;
78 WS_MINEFF DATE;
79 WS_MAXEFF DATE;
80 WS_MINMOD DATE;
81 WS_MAXMOD DATE;
82 WS_POLEFF DATE;
83 WS_EXPOEFF DATE;
84 WS_EXPOEFF2 DATE;
85 WS_ANNDATE DATE;
86 WS_REPORTDATE DATE;
87 VAR_BUREAU USOG.USOG_WC_ORG_MAIN%TYPE;
88 VAR_BATCH VARCHAR2(1);
89 VAR_DATE_IN DATE;
90 VAR_POLICY_IN RATP.POLICY_NUM%TYPE;
91 VAR_START VARCHAR2(8);
92 VAR_STOP VARCHAR2(8);
93 WS_USE_1111 unex.UNEX_WC_RECORD_TYPE%type := 'N';
94 WS_XMIT unex.UNEX_XMIT_IND%type;
95 WS_CLAIM_COUNT NUMBER;
96
97 ws_sqlcode number;
98 --
99 CURSOR GET_POLICY
100 IS
101 SELECT DISTINCT UNPL.UNPL_WC_NO AS UNPLID,
102 TUNX.POLICY_NUMBER AS POL_1 ,
103 UNPL.UNPL_WC_POLY_EFF_DATE AS POLEFF,
104 TUNX.ANNIV_DTE AS ANNDATE
105 FROM TEMP_UNIT_EXPO TUNX,
106 UNIT_STATS_POLICY UNPL
107 WHERE TUNX.POLICY_NUMBER = UNPL.UNPL_WC_POLY_NUMBER
108 AND UNPL.UNPL_WC_REPORT_DATE = WS_REPORTDATE
109 AND SUBSTR (TUNX.CLASS_CODE, 1, 1) BETWEEN ('0') AND ('9')
110 AND TUNX.STATE_NUMBER = WS_STATE_NO
111 AND MONTHS_BETWEEN (WS_REPORTDATE, LAST_DAY

```

```

112     (UNPL.UNPL_WC_POLY_EFF_DATE)) = 18
113     ORDER BY TUNX.POLICY_NUMBER DESC ;
114 --
115 CURSOR DEVIATED_PREM
116 IS
117     SELECT DISTINCT TUNX.POLICY_NUMBER AS DPOL_NUM ,
118         NVL (TUNX.MULTIPLIER, 0)      AS RATE,
119         NVL(tunx.modifier_value,0)    AS DEVIATION,
120         SUM (NET_PREMIUM)             AS NetStdPx,
121         SUM (tunx.standard_premium)   AS StdPx,
122         CLASS_CODE                     AS clscde
123 FROM TEMP_UNIT_EXPO TUNX
124 WHERE SUBSTR (CLASS_CODE, 1, 4) IN
125     ( SELECT DISTINCT SUBSTR (CLASS_CODE, 1, 4) AS CLASSCODE
126   FROM STCL
127   WHERE USE_MOD_IND = 'Y'
128   AND CAT_CODE   IN ('MANUAL', 'MODFAC', 'INCLIM', 'MERIT'))
129 AND SUBSTR(CLASS_CODE, 1, 4) IN
130     (SELECT rade.class_code FROM rade WHERE NVL(DEVIATION_PCT,1) IS NOT NULL)
131 AND SUBSTR (CLASS_CODE, 1, 4) NOT IN ('9807', '9812','6199')
132 AND TUNX.POLICY_NUMBER      = WS_POLICY
133 AND TUNX.STATE_NUMBER      = WS_STATE_NO
134 GROUP BY
135     TUNX.POLICY_NUMBER ,
136     NVL (TUNX.MULTIPLIER, 0) ,
137     NVL(tunx.modifier_value,0) ,
138     CLASS_CODE
139 ORDER BY
140     TUNX.POLICY_NUMBER ;
141 --
142 CURSOR FIND_STATES
143 IS
144     SELECT STAT_GB_STATE_NO STATE_NO FROM STAT ORDER BY 1 DESC;
145 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
146 -- SELECT LOGICALS
147 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
148 BEGIN
149     WS_SECTION := 'WCUNEX2_00A';
150     SELECT UPPER (UETB.UETB_DP_VALUE) INPUT_SRC
151     INTO VAR_BATCH
152     FROM USER_EXIT_TABLE UETB
153     WHERE UETB.UETB_DP_FIELD_NAME   = 'RPT_OPT'
154           AND UETB.UETB_DP_PROGRAM_NAME = 'WCUNEX_STATS'
155           AND UETB.UETB_DP_USER_NAME  = USER;
156 --
157     WS_SECTION := 'WCUNEX2_00B';
158     SELECT UPPER (UETB.UETB_DP_VALUE) REPORT_DTE
159     INTO VAR_DATE_IN
160     FROM USER_EXIT_TABLE UETB
161     WHERE UETB.UETB_DP_FIELD_NAME   = 'ENDDATE'
162           AND UETB.UETB_DP_PROGRAM_NAME = 'WCUNEX_STATS'
163           AND UETB.UETB_DP_USER_NAME  = USER;
164 --
165     WS_SECTION := 'WCUNEX2_00C';
166     SELECT UPPER (UETB.UETB_DP_VALUE) POL_NUMBER
167     INTO VAR_POLICY_IN

```

```

167         FROM USER_EXIT_TABLE UETB
168         WHERE UETB.UETB_DP_FIELD_NAME    = 'DATAOPT'
169             AND UETB.UETB_DP_PROGRAM_NAME = 'WCUNEX_STATS'
170             AND UETB.UETB_DP_USER_NAME   = USER;
171         --
172         SELECT TO_CHAR (SYSDATE, 'HH24:MI:SS') INTO VAR_START FROM DUAL;
173
174
175 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
176 -- MAIN PROGRAM
177 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
178 WS_REPORTDATE := VAR_DATE_IN;
179 FOR D1      IN FIND_STATES
180 LOOP
181     WS_UNPL_ID := NULL;
182     WS_POLICY  := NULL;
183     WS_ANNDATE := NULL;
184     WS_CLASS   := NULL;
185     WS_PAYROLL := NULL;
186     WS_SECTION := 'WCUNEX2_01';
187     WS_STATE_NO := D1.STATE_NO;
188     --
189     FOR A1 IN GET_POLICY
190     LOOP
191         WS_SECTION := 'WCUNEX2_02';
192         WS_UNPL_ID := A1.UNPLID;
193         WS_POLICY  := A1.POL_1;
194         WS_ANNDATE := A1.ANNDATE;
195         WS_POLEFF  := A1.POLEFF;
196         -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
197         -- PLI32502.18116
198         -- DETERMINE deviated premium for Massachusetts
199         -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
200
201
202         WCX.GET_DEVIATED(WS_UNPL_ID,WS_POLEFF,WS_POLICY,WS_STATE_NO,WS_REPOR
203         TDATE);
204
205
206 -- *****
207 -- CLASS CODE 1111 SECTION
208 -- *****
209 BEGIN
210     WS_SECTION  := 'WCUNEX2_03';
211     WS_CLASS    := NULL;
212     WS_CLAIM_COUNT := 0;
213     WS_PAYROLL  := 0;
214     -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
215     -- DETERMINE IF THE PAYROLL CLASS CODES AND/OR THE TERRORISM CLASS CODES
216     -- (SINCE THEY ARE DETERMINED BY THE PAYROLL) HAVE DOLLAR AMOUNTS
217     ASSIGNED
218     -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
219     SELECT TUNX.POLICY_NUMBER POLNUM,
220            SUM (TUNX.PAYROLL_AMOUNT) TTL_PAYROLL

```

```

220 INTO WS_POLICY,
221     WS_PAYROLL
222 FROM TEMP_UNIT_EXPO TUNX
223 WHERE TUNX.POLICY_NUMBER = WS_POLICY
224 AND TUNX.STATE_NUMBER = WS_STATE_NO
225 AND SUBSTR (CLASS_CODE, 1, 4) IN
226     (SELECT DISTINCT SUBSTR (CLASS_CODE, 1, 4) AS CLASSCODE
227     FROM STCL
228     WHERE CAT_CODE IN ('MANUAL', 'TERROR')
229     )
230 HAVING SUM (TUNX.PAYROLL_AMOUNT) = 0
231 GROUP BY TUNX.POLICY_NUMBER
232 ORDER BY TUNX.POLICY_NUMBER;
233 EXCEPTION
234 WHEN NO_DATA_FOUND THEN
235     WS_PAYROLL := 1;
236 END;
237 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
238 -- DETERMINE IF THERE ARE ANY CLAIMS ATTACHED TO THE POLICY
239 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
240 WS_SECTION := 'WCUNEX2_04';
241 SELECT COUNT ( *) CLAIM_COUNT
242 INTO WS_CLAIM_COUNT
243 FROM UNIT_STATS_LOSS UNLS
244 WHERE UNLS.UNLS_UNPL_WC_NO IN
245     (SELECT UNPL.UNPL_WC_NO
246     FROM UNIT_STATS_POLICY UNPL
247     WHERE UNPL.UNPL_WC_POLY_NUMBER = WS_POLICY
248     )
249 AND UNLS.UNLS_STAT_GB_STATE_NO = WS_STATE_NO;
250 WS_SECTION := 'WCUNEX2_05';
251 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
252 -- SET ALL VALUES TO ZERO AND INSERT THE 1111 CLASS CODE IF THERE ARE NO
253 -- CLAIMS AND NO PAYROLL FOR THE POLICY
254 -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
255 IF WS_CLAIM_COUNT = 0 THEN
256     IF WS_PAYROLL = 0 THEN
257         --
258         IF WS_STATE_NO NOT IN ('04', '29', '37') THEN
259             SELECT UNIT_STATS_EXPOSURE_SEQUENCE.NEXTVAL INTO WS_UNEX_WC_NO
260             FROM DUAL;
261             WS_CLASS := '1111';
262             WS_MODPX := 0;
263             WS_NETPX := 0;
264             WS_MODVAL := 0;
265             WS_PAYROLL := 0;
266             WS_SPLITNO := 0;
267             WS_BURT := 0;
268             WS_MODEFF := WS_POLEFF;
269             WS_BURT_EFF := WS_POLEFF;
270             WS_AFTERMOD := 'E';
271             WS_XMIT := 'Y';
272             WS_SECTION := 'WCUNEX2_06';
273             WCX.TEMP_INSERT_UNEX (WS_UNEX_WC_NO, WS_UNPL_ID, WS_REPORTDATE,
274             WS_STATE_NO, WS_CLASS, WS_MODVAL, WS_MODEFF, WS_BURT_EFF,
275             WS_PAYROLL, WS_NETPX, WS_MODPX, WS_SPLITNO, WS_BURT, WS_AFTERMOD,

```

```

273     WS_XMIT, ws_sqlcode) ;
274     COMMIT;
275 ELSE
276     WS_SECTION := 'WCUNEX2_07';
277     DBMS_OUTPUT.PUT_LINE (WS_POLICY||': CLASS CODE 1111 NOT APPLICABLE IN
278     NEW JERSEY, CALIFORNIA, or PENNSYLVANIA');
279     --PLI32502.21102
280     SELECT UNIT_STATS_EXPOSURE_SEQUENCE.NEXTVAL
281     INTO WS_UNEX_WC_NO
282     FROM DUAL;
283     --
284     BEGIN
285         SELECT TUNX.CLASS_CODE      AS CLASS_CODE,
286                SUM (TUNX.STANDARD_PREMIUM) AS STDPX_TTL
287         INTO WS_CLASS,
288                WS_STDPX
289         FROM TEMP_UNIT_EXPO TUNX
290         WHERE TUNX.POLICY_NUMBER    = WS_POLICY
291         AND SUBSTR (CLASS_CODE, 1, 4) IN
292             (SELECT DISTINCT SUBSTR (CLASS_CODE, 1, 4) AS CLASSCODE
293              FROM STCL
294              WHERE CAT_CODE IN ('MANUAL', 'TERROR')
295              )
296         AND STANDARD_PREMIUM IN
297             (SELECT MAX (TUNX_A.STANDARD_PREMIUM)
298              FROM TEMP_UNIT_EXPO TUNX_A
299              WHERE TUNX_A.POLICY_NUMBER = WS_POLICY
300              )
301         GROUP BY TUNX.POLICY_NUMBER,
302                TUNX.CLASS_CODE
303         ORDER BY TUNX.POLICY_NUMBER;
304     EXCEPTION
305     WHEN NO_DATA_FOUND THEN
306         EXIT;
307     END;
308     --
309     WS_SECTION := 'WCUNEX2_08';
310     WS_MODPX   := 0;
311     WS_NETPX    := 0;
312     WS_MODVAL   := 0;
313     WS_PAYROLL  := 0;
314     WS_SPLITNO  := 0;
315     WS_BURT     := 0;
316     WS_MODEFF   := WS_POLEFF;
317     WS_BURT_EFF := WS_POLEFF;
318     WS_AFTERMOD := 'E';
319     WS_XMIT     := 'Y';
320     WCX.TEMP_INSERT_UNEX (WS_UNEX_WC_NO, WS_UNPL_ID, WS_REPORTDATE,
321     WS_STATE_NO, WS_CLASS, WS_MODVAL, WS_MODEFF, WS_BURT_EFF,
322     WS_PAYROLL, WS_NETPX, WS_MODPX, WS_SPLITNO, WS_BURT, WS_AFTERMOD,
323     WS_XMIT, ws_sqlcode) ;
324     COMMIT;
325     dbms_output.put_line(WS_SECTION||' '||WS_PAYROLL||' '||WS_STATE_NO);
326     --
327 END IF;
328 END IF;

```

```

324     END IF;
325     END LOOP;
326 END LOOP; -- FIND STATES
327     -- (=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)
328     SELECT TO_CHAR (SYSDATE, 'HH24:MI:SS')
329     INTO VAR_STOP
330     FROM DUAL;
331
332     DBMS_OUTPUT.PUT_LINE (VAR_START);
333     DBMS_OUTPUT.PUT_LINE (VAR_STOP);
334     --
335 EXCEPTION
336 WHEN NO_DATA_FOUND THEN
337     DBMS_OUTPUT.PUT_LINE ('***** NO DATA FOUND *****');
338     DBMS_OUTPUT.PUT_LINE ('SECTION'||WS_SECTION);
339     DBMS_OUTPUT.PUT_LINE ('SQLCODE: '||SQLCODE||' POLICY: '||WS_POLICY);
340     DBMS_OUTPUT.PUT_LINE ('UNEX ID: '||WS_UNEX_WC_NO||' UNPL ID: '||WS_UNPL_ID||'
REPORT DATE: '||WS_REPORTDATE);
341     DBMS_OUTPUT.PUT_LINE ('STATE: '||WS_STATE_NO||' CLASS: '||WS_CLASS||' NET:
'||WS_NETPX||' MOD: '||WS_MODPX||' SPLIT: '||WS_SPLITNO||' TYPE: '||WS_AFTERMOD);
342     DBMS_OUTPUT.PUT_LINE (SQLERRM);
343     DBMS_OUTPUT.PUT_LINE (DBMS_UTILITY.FORMAT_ERROR_BACKTRACE);
344     DBMS_OUTPUT.PUT_LINE ('***** END OF LINE *****');
345 WHEN TOO_MANY_ROWS THEN
346     DBMS_OUTPUT.PUT_LINE ('***** TOO MANY ROWS *****');
347     DBMS_OUTPUT.PUT_LINE ('SECTION'||WS_SECTION);
348     DBMS_OUTPUT.PUT_LINE ('SQLCODE: '||SQLCODE||' POLICY: '||WS_POLICY);
349     DBMS_OUTPUT.PUT_LINE ('UNEX ID: '||WS_UNEX_WC_NO||' UNPL ID: '||WS_UNPL_ID||'
REPORT DATE: '||WS_REPORTDATE);
350     DBMS_OUTPUT.PUT_LINE ('STATE: '||WS_STATE_NO||' CLASS: '||WS_CLASS||' NET:
'||WS_NETPX||' MOD: '||WS_MODPX||' SPLIT: '||WS_SPLITNO||' TYPE: '||WS_AFTERMOD);
351     DBMS_OUTPUT.PUT_LINE (SQLERRM);
352     DBMS_OUTPUT.PUT_LINE (DBMS_UTILITY.FORMAT_ERROR_BACKTRACE);
353     DBMS_OUTPUT.PUT_LINE ('***** END OF LINE *****');
354 WHEN OTHERS THEN
355     DBMS_OUTPUT.PUT_LINE ('***** ERROR *****');
356     DBMS_OUTPUT.PUT_LINE ('SECTION'||WS_SECTION);
357     DBMS_OUTPUT.PUT_LINE ('SQLCODE: '||SQLCODE||' POLICY: '||WS_POLICY);
358     DBMS_OUTPUT.PUT_LINE ('UNEX ID: '||WS_UNEX_WC_NO||' UNPL ID: '||WS_UNPL_ID||'
REPORT DATE: '||WS_REPORTDATE);
359     DBMS_OUTPUT.PUT_LINE ('STATE: '||WS_STATE_NO||' CLASS: '||WS_CLASS||' NET:
'||WS_NETPX||' MOD: '||WS_MODPX||' SPLIT: '||WS_SPLITNO||' TYPE: '||WS_AFTERMOD);
360     DBMS_OUTPUT.PUT_LINE (SQLERRM);
361     DBMS_OUTPUT.PUT_LINE (DBMS_UTILITY.FORMAT_ERROR_BACKTRACE);
362     DBMS_OUTPUT.PUT_LINE ('***** END OF LINE *****');
363 END;
364 /
365
366
367

```