

IDENTIFICATION DIVISION.
PROGRAM-ID. WCLOSS_STATS.
AUTHOR. STEVE CANCELLI.
DATE-WRITTEN. 03/18/90.

-----*

* PROGRAM NAME : WCLOSS_STATS *

* FUNCTION : THIS PROGRAM PROVIDES A REPORT LISTING WORK COMP *
* POLICIES THAT HAVE AN EXPIRATION DATE WITHIN *
* THE MONTH WHICH IS SIX MONTHS PRIOR TO THE CLAIMS *
* DATE CONTROL VALUES(OR ANY SUBSEQUENT YEARS). *
* ALSO THESE W/C POLICIES MUST HAVE EITHER OPEN *
* CLAIMS OR CLAIMS WHICH HAVE BEEN CLOSED/REOPENED *
* SINCE THE LAST UNIT STATS WERE REPORTED. *

* INPUT : THE INPUT DATA FOR THIS PROGRAM IS RETRIEVED FROM *
* THE FOLLOWING DATA BASE TABLES AND COLUMNS: *

- * 1. DATE_CONTROL - DTCL_GB_END_DATE. *
- * 2. LOSS_RESERVE - LSRV_CL_RESV_AMOUNT. *
- * 3. LOSS_PAID - LSPD_CL_PAID_AMOUNT. *
- * 4. RESV_HIST - RVHS_BSOR_GB_ACCOUNT, *
- * RVHS_GB_POL_NUMBER, *
- * RVHS_GB_CLASS_CODE, *
- * RVHS_GB_STATE, *
- * RVHS_CL_OCCURRENCE_DATE, *
- * RVHS_CL_CLAIMN_NO, *
- * RVHS_RSTP_CL_TYPE_CODE, *
- * RVHS_CLIC_CL_CODE, *
- * RVHS_CL_CLOSE_DATE, *
- * RVHS_CL_REOPEN_DATE, *
- * RVHS_CL_NO, *
- * RVHS_RSTP_CL_TYPE_CODE. *

* OUTPUT : THE REPORT GENERATED BY WCLOSS_STATS.PCO *
* CONSISTS OF WORK COMP UNIT STATISTICS. *
* EACH MEDICAL AND IN INDEMNITY CLAIM IS *
* DETAILED AND ORDERED BY STATE WITH *
* THE CURRENT RESERVE, PAID AMOUNT, AND THE *
* TOTAL INCURRED LISTED FOR EACH. *
* TOTALS ARE PROVIDED FOR NUMBER OF CLAIMS OF EACH *
* TYPE AND THE CURRENT RESERVE, PAID AMOUNT, *
* AND THE TOTAL INCURRED FOR EACH STATE. *
* TOTALS ARE ALSO PROVIDED AT THE END OF THE REPORT *
* WITH THE SUMS OF THE TOTALS FOR EACH STATE. *

* IT ALSO WRITES OUT A DATA FILE THAT IS LOADED *
* INTO A PC APPLICATION THAT CREATES THE UNIT STAT *
* REPORTS. *

* TABLES : THE FOLLOWING TABLES ARE ACCESSED WITH NO *
* UPDATES: *
* DATE_CONTROL, *
* LSRV, *
* LOSS_PAID, *

```

57 * STATE, *
58 * THE FOLLOWING TABLES ARE ACCESSED WITH *
59 * UPDATES OCCURRING: *
60 * RESV_HIST, *
61 * *
62 * LOGICALS : DATE$OPT - SIGNALS THE INPUT OF AN END$DATE *
63 * IF DATE$OPT IS NOT 'Y' THEN THE DATE *
64 * IS RETRIEVED FROM DATE CONTROL. *
65 * END$DATE - THE LATEST DATE TO BE REPORTED ON *
66 * THE REPORT WILL INCLUDE DATA FOR *
67 * POLICIES WITH EXPIRATION DATES OF *
68 * 6 MONTHS PRIOR TO THE DATE ENTERED. *
69 * *
70 * *
71 * ----- *
72 * ----- *
73 * MODIFICATIONS *
74 * ----- *
75 * ID-NBR INI. DATE DESCRIPTION *
76 * *
77 * SGC 01-01-89 INITIAL CREATION OF PROGRAM. *
78 * *
79 * NR7390 KGM 09-12-90 ADDED LOSS RVHS_WCCC_TYPE_CODE, *
80 * WORK COMP COVERAGE CODE TO REPORT *
81 * *
82 * NR7390 SGC 06-NOV-90 ADDED THE LOGIC TO CREATE INDEPENDENT *
83 * REPORTS FOR UP TO 6 STATES. *
84 * THESE INDIVIDUAL STATE REPORTS ARE *
85 * IN THE OUTPUT FILE *
86 * WCLOSS_STATS_STATE.PRT. THIS REPORT *
87 * IS IDENTICAL TO THE ORIGINAL. *
88 * *
89 * NR4090 12-MAR-91 RWW - ADDED CODE TO DO A TO_DATE FUNCTION WITH *
90 * THE FORMAT DD-MON-YY ON SYSDATE. THE *
91 * REASON IS BECAUSE SYSDATE INCLUDES THE *
92 * TIME UNLESS A TO_DATE FUNCTION IS *
93 * PERFORMED ON IT. *
94 * *
95 * NR791 08-APR-91 AKS - ADDED CODE TO *
96 * 'FSI' POLICIES FROM BEING PICKED UP. *
97 * *
98 * PL29591I 07-JAN-92 MJR - ADDED CODE TO DRIVING CURSOR TO INCLUDE *
99 * REOPENED CLAIMS. *
100 * *
101 * GG *
102 * NR1091 MJR 25-MAR-92 - MODIFY PROGRAM FOR SEGREGATION OF PAIDS *
103 * *
104 * NR1388 (DLG) 30-NOV-92 - PROVIDE WORK COMP DATA FROM NCCI *
105 * ADDED OUTPUT OF A POLICY DATA FILE, *
106 * LOCATION DATA FILE, AND CLAIM DATA FILE *
107 * *
108 * NR1388 KGM 31-DEC-92 CHANGED SETTING OF CLAIM STATUS SO THAT *
109 * IN NNCCI FILE ALL CLAIMS ARE EITHER *
110 * OPEN OR RE-OPEN. *
111 * ALSO DROPPED LEADING W FROM CLAIM NUMBER*
112 * *

```

113	*	NR1388	KGM	15-JAN-93	WHEN POL PREFIX DOES NOT START WITH	*
114	*				W ADDED 4030 TO GET PLSM PREFIX.	*
115	*				CHANGED CLAIM NUMBER IN AGO_CLAM FILE	*
116	*				TO DROP LEADING W FROM CLAIM NUMBER	*
117	*					
118	*	NR1388	KGM	24-APR-93	NUMEROUS CHANGES FOR UNIT-STAT	
119	*				REPORTING TO AGO PC SYSTEM - ADDITION	
120	*				OF SSN/FED. ID TO REPORT	
121	*					
122	*	NR1388	KGM	31-AUG-93	CORRECTION IN DETERMINATION OF CLAIM	
123	*				STATUS;	
124	*					
125	*	NR1388	KGM	17-SEP-93	REMOVED REQUIREMENT THAT A POLICY HAVE A	
126	*				COMPLETED AUDIT BEFORE IT CAN BE ADDED TO	
127	*				THE DATA FILE. ALSO REMOVED THE SELECT	
128	*				OF THE BURT RATE - THE BURT RATE WAS NOT	
129	*				BEING USED	
130	*					
131	*	NR1388	KGM	29-JUL-94	ADDED NEW FIELDS TO NCCI RECORD. THE	
132	*				NEW FIELDS (EXCEPT SSN) ARE NOT BEING	
133	*				USED AT THIS TIME.	
134	*					
135	*	NR1388	KGM	23-SEP-94	MODIFIED TO ADD DATA REQUIRED FOR NEW	
136	*				250 CHARACTER FORMAT NCCI RECORD.	
137	*					
138	*	PLI27495	CAH	30-SEP-97	CHANGES TO POPULATE UNIT_STATS_LOSS	
139	*				TABLE FOR ELECTRONIC FILING OF	
140	*				UNIT STATISTICS TO COMPLY WITH ASWG	
141	*				FILE FORMAT (250 CHAR FORMAT).	
142	*					
143	*	PLI27495	MJRIO	27-OCT-97	ADDED CODE FOR Y2K COMPLIANCE.	
144	*				MADE VARIOUS CHANGES AND CORRECTIONS	
145	*				ACCORDING TO NCCI REGULATIONS (CAH).	
146	*					
147	*	PLI27495	CAH	09-FEB-98	CHANGED PRINT LINES FROM 140 TO 176.	
148	*				SSN WAS GETTING CHOPPED OFF FROM PRIOR	
149	*				CHANGE.	
150	*					
151	*	PLI7498	CAH	27-AUG-98	RECOMPILED TO FIX ERROR CAUSED MOST	
152	*				LIKELY BY UPGRADE FROM ORACLE 7.1 TO 7.3.	
153	*				COMMENTED OUT DISPLAY OF RVHS_CL_NO	
154	*					
155	*	PLI7498	CAH	14-JAN-99	COMMENTED OUT CODE THAT POPULATED THE	
156	*				UNLS_CL_PREV_REP_IND AND UNLS_CL_UPDATE_TYPE.	
157	*				POPULATING THEM IN GBFILELAY INSTEAD.	
158	*					
159	*	PLI17498	CAH	16-FEB-99	ROUNDED SUMS OF INDEMNITY AND MEDICAL PAYMENTS	
160	*				TO THE NEAREST DOLLAR - PER BOB POTTER.	
161	*					
162	*	PLI7498	CAH	04-MAR-99	COMMENTED OUT CALL TO 4040-GET-CLIC-CODE.	
163	*				THIS PARAGRAPH WAS OVER WRITING THE CODE	
164	*				FETCHED, REPLACING IT WITH AN INJURY CODE	
165	*				NOT NECESSARILY BASED ON THE RIGHT RVHS ROW.	
166	*				THIS ERROR ONLY OCCURED ON THE REPORT, IS NOT	
167	*				PRESENT IN THE UNLS TABLE.	
168	*					

169 * PLI1801 DDG 01-AUG-2001 ADDED LOGIC TO EXTRACT ADDITIONAL INFO
170 * WANTED BY CALIFORNIA
171 *
172 * PLI30579 DDG 05-MAY-2003 GUI CONVERSION
173 *
174 * PLI31530.3248 JPC 28-DEC-2004 MAKE CHANGES TO ALLOW FOR FIGURING
175 * AND REPORTING GROSS/NET AMOUNTS W/DEDUCTIBLES
176 *
177 * PLI31929.4654 JPC 07-JAN-2005 Correct logic that selects loss type codes from WCCC
178 *
179 * PLI31990.4791 JPC 23-MAR-2005 Modify logic to select policies for 6th - 10th reports
180 *
181 * PLI31990.5560 JPC 23-NOV-2005 Made deductible code SELECT more dynamic; also
changed
182 * number of months selected from 67 to 127
183 *
184 * PLI32439.6164 DDG 31-AUG-2006 FIX SEVERAL ISSUES - REHAB INDICATOR, CLPT
INFO ASSOCIATED
185 * WITH MOST RECENTLY CREATED RESERVE
186 *
187 * PLI32502.7122 JPC 10-JAN-2007 ADD HANDLING FOR NJ INJURY CODE '09'
188 *
189 * PLI32511.6660 DDG 06-JUN-2007 CHANGE TO USE TOTALED VOC REHAB IF AMT IN
RESERVE IS LESS
190 * CHANGE TO USE MOST RECENT 'WI' RESERVE FOR VOC REHAB
191 *
192 * PLI32502.9352 JPC 12-SEP-2007 REMOVED OLD LOGIC FOR DETERMINING NCCI vs.
NON-NCCI STATES
193 *
194 * PLI32502.9994 JPC 29-NOV-2007 CORRECTED CALCULATION FOR RESERVES WITH
DEDUCTIBLES
195 *
196 * PLI32502.9871 JPC 17-JAN-2008 modified logic to use new columns from cloc/rvhs;also
added column for hire date
197 *
198 * PLI32502.10620 JPC 07-FEB-2008 ADDED MOVE FROM RVHS VARIABLES TO UNLS
VARIABLES FOR RERUN CURSOR;
199 * REMOVED POPULATION OF HIRE DATE
200 *
201 * PLI32502.10826 JPC 02-APR-2008 MADE INCURRED ALAE = PAID ALAE; SET NUMBER
OF CLAIMS VALUE TO 1
202 * PLI32502.11013 JPC 02-APR-2008 REMOVED TIE TO CLPT IN PAID VOC REHAB
CALCULATION IN 1800-GET-CA-FIELDS
203 * PLI32502.11196 JPC 02-APR-2008 ADDED PARAGRAPH 3010-ULCT-DECODE TO
HANDLE CLAIM DETAIL CODES
204 *
205 * PLI32502.11381 JPC 07-MAY-2008 CORRECTED LOGIC FOR CLAIM DETAIL CODES TO
ACCEPT NEW CODES OR USE
206 * EXISTING CODES IN THEIR ABSENCE
207 * PLI32502.11531 JPC 07-MAY-2008 COMMENTED OUT EQUATION THAT ADDS THE
DEDUCIBLE TO THE RESERVE
208 *
209 * PLI32502.11429 JPC 14-MAY-2008 CORRECT FORMULA FOR CALIFORNIA GROSS
INCURRED
210 * PLI32502.11594 JPC 14-MAY-2008 REFINE MAIN SELECT SO POLICIES ARE NOT
REPORTED TOO SOON.

```

211 *
212 * PLI32502.11843 JPC 18-JUN-2008 added logic to populate
UNLS_CL_MANG_CARE_ORG_TYPE based on related UNPL value
213 * PLI32502.11850 JPC 18-JUN-2008 zero out ALAE totals when they are negative
214 *
215 * PLI32502.12846 JPC 28-OCT-2008 allow SSN in California reports
216 *
217 * PLI32502.11597 JPC 19-NOV-2008 make LSPD selections more generic to allow for
multiple RVHS ID's for one claim
218 *
219 * PLI32502.13069 JPC 04-DEC-2008 added reserve type codes as part of LSPD selection
criteria
220 *
221 * PLI32502.14281 JPC 08-APR-2009 CORRECT COMPUTATION ERROR ON BALANCING
LOGIC
222 *
223 * PLI32502.14568 JPC 07-MAY-2009 CHANGE DEFAULT VALUE OF
WS-UNLS-LOSS-COVG TO '01'
224 *
225 * PLI32502.15425 JPC 04-AUG-2009 ADD LOGIC CURSORS THAT WILL HANDLE
DOUBLE-DIGIT CLIC CODES
226 *
227 * PLI32502.15572 JPC 01-SEP-2009 Include negative amounts on exception report.
228 *
229 * PLI32502.9408 JPC 14-OCT-2009 Change status marker for reopened first reports
230 *
231 * PLI32502.16006 JPC 20-NOV-2009 Add call to procedure P_ID_LATE_CLAIMS
232 *
233 * PLI32502.16651 JPC 17-FEB-2010 Add logic to generate catastrophe numbers
234 *
235 * PLI32502.18292 JPC 18-AUG-2010 Calculate attorney fees
236 *
237 * PLI32502.20137 JPC 18-JAN-2011 Correct logic for catastrophe numbers
238 *
239 * PLI32502.20196 JPC 21-JAN-2011 Add gunshot (GUNS) as a cause of loss
240 *
241 * PLI32502.19083 JPC 03-FEB-2011 Only use HMO indicator when not in California
242 *
243 * PLI32502.21890 JPC 12-OCT-2011 Set default Catastrophe Number to zero
244 * PLI32502.22151 JPC 29-NOV-2011 Remove NJ logic from PLI32502.7122
245 *
246 * PLI32502.23637 JPC 09-AUG-2012 Make legal fees conditional
247 *
248 * PLI32502.27332 JPC Add ABS around WS-UNLS-CL-PAID-VOC-REHAB
249 * -----*
250 * -----*
251 * ENVIRONMENT DIVISION *
252 * -----*
253 *
254 ENVIRONMENT DIVISION.
255 CONFIGURATION SECTION.
256 *
257 SOURCE-COMPUTER. VAX-8350.
258 OBJECT-COMPUTER. VAX-8350.
259 *
260 SPECIAL-NAMES.

```

```

261         ARGUMENT-NUMBER IS AUG-NUM
262         ARGUMENT-VALUE IS AUG-VAL.
263     *
264     INPUT-OUTPUT SECTION.
265     FILE-CONTROL.
266     *
267     *   SELECT REPORT-OUT ASSIGN TO "APPS$RPT:WCLOSS_STATS.PRT".
268     *   SELECT REPORT-OUT ASSIGN TO REPORT-FILENAME.
269
270     *   SELECT REPORT-OUT2 ASSIGN TO "APPS$RPT:WCLOSS_STATS_STATE.PRT".
271     *   SELECT REPORT-OUT2 ASSIGN TO REPORT-FILENAME1.
272
273     *   SELECT CLAIM   ASSIGN TO "APPS$RPT:AGO_CLAM.DAT".
274     *   SELECT CLAIM   ASSIGN TO CLAIM-REPORT-FILE.
275
276     *   SELECT REPORT-OUT3 ASSIGN TO "APPS$RPT:WCLOSS_STATS_EXCEPTION.PRT".
277     *   SELECT REPORT-OUT3 ASSIGN TO REPORT-FILENAME2.
278
279     *-----*
280     *           DATA DIVISION           *
281     *-----*
282     DATA DIVISION.
283     FILE SECTION.
284     *
285     FD REPORT-OUT.
286     *
287     01 PRINT-LINE          PIC X(176).
288     *
289     FD REPORT-OUT2.
290     *
291     01 PRINT-LINE2        PIC X(176).
292     *
293     FD REPORT-OUT3.
294     *
295     01 PRINT-LINE3        PIC X(176).
296     *
297     *=====
298     *   CLAIM RECORD LAYOUT
299     *-----
300     FD CLAIM.
301     01 CLAIM-OUTPUT.
302     02 CLAIM-RECORD-LAYOUT.
303     05 CLAIM-FILLER        PIC X(303).
304     02 CLAIM_REC REDEFINES CLAIM-RECORD-LAYOUT.
305     05 CLAIM-POLICY-NUMBER PIC X(16).
306     05 CLAIM-CUSTOMER-NUMBER.
307     10 CLAIM-CUSTOMER-NUMBER-F4 PIC S9(04).
308     10 CLAIM-CUSTOMER-NUMBER-L1 PIC Z.
309     05 FILLER              PIC X(05).
310     05 CLAIM-INCEPTION-DATE PIC S9(08).
311     05 CLAIM-NUMBER        PIC X(12).
312     05 CLAIM-AVERAGE-WEEKLY-WAGE PIC S9(05).
313     05 CLAIM-BODY-PART-CODE PIC X(04).
314     05 CLAIM-NATURE-OF-INJURY-CODE PIC X(04).
315     05 CLAIM-CAUSE-OF-INJURY-CODE PIC X(04).
316     05 CLAIM-LOSS-CODE-TYPE PIC X(04).

```



```

317      05 CLAIM-INJURY-CODE          PIC S9(02).
318      05 CLAIM-SOURCE-OF-INJURY     PIC X(04).
319  *
320      05 CLAIM-ACCIDENT-DATE         PIC S9(08).
321      05 CLAIM-ORIGINAL-CLAIM-DATE   PIC S9(08).
322      05 CLAIM-STATUS                PIC X(01).
323      05 CLAIM-TRANSACTION-DATE      PIC S9(08).
324      05 CLAIM-CATASTROPHE-NUMBER    PIC S9(02).
325      05 CLAIM-CLASS-CODE            PIC X(07).
326      05 CLAIM-AGENT-CODE            PIC X(04).
327      05 CLAIM-INDEMNITY-RESERVE-AMT PIC S9(07)V99.
328      05 CLAIM-INDEMNITY-PAYMENT-AMT PIC S9(07)V99.
329      05 CLAIM-MEDICAL-RESERVE-AMT   PIC S9(07)V99.
330      05 CLAIM-MEDICAL-PAYMENT-AMT   PIC S9(07)V99.
331  *
332      05 CLAIM-REHAB-RESERVE-AMT     PIC S9(07)V99.
333      05 CLAIM-REHAB-PAYMENT-AMT     PIC S9(07)V99.
334      05 CLAIM-EXPENSE-RESERVE-AMT   PIC S9(07)V99.
335      05 CLAIM-EXPENSE-PAYMENT-AMT   PIC S9(07)V99.
336      05 CLAIM-EXPENSE-CODE          PIC X(03).
337      05 CLAIM-LOCATION-NUMBER         PIC S9(02).
338      05 CLAIM-LOSS-COVERAGE-CODE    PIC X(02).
339      05 CLAIM-POLNUM-DATE-TIME-STAMP.
340      10 CLAIM-POLNUM-POLNO          PIC X(16).
341      10 CLAIM-POLNUM-DT-TIME.
342      15 CLAIM-POLNUM-DATE           PIC X(10).
343      15 CLAIM-POLNUM-TIME           PIC X(08).
344      05 CLAIM-STATE                 PIC X(02).
345      05 CLAIM-SSN                   PIC X(09).
346      05 CLAIM-DEDUCT-TYPE           PIC X(04).
347      05 CLAIM-DEDUCT-PCT            PIC S9(03) COMP-3.
348      05 CLAIM-HMO-TYPE              PIC X(02).
349      05 CLAIM-OCCUP-DESC            PIC X(18).
350      05 CLAIM-VOC-REHAB-IND         PIC X(01).
351      05 CLAIM-LUMP-SUM-IND          PIC X(01).
352      05 CLAIM-FRAUD-IND             PIC X(02).
353      05 CLAIM-DEDUCT-CODE           PIC X(02).
354      05 CLAIM-CLAIMMAINT-ATT-FEE    PIC S9(07)V99.
355      05 CLAIM-EMPLOYER-ATT-FEE      PIC S9(07)V99.
356      05 CLAIM-ALLOC-LAE-PD          PIC S9(07)V99.
357      05 CLAIM-ALLOC-LAE-IN          PIC S9(07)V99.
358
359  *=====
360  *====
361  *-----*
362  *          WORKING STORAGE SECTION          *
363  *-----*
364  WORKING-STORAGE SECTION.
365  *-----*
366  *          SQL WORK FIELD AREA              *
367  *-----*
368  EXEC SQL BEGIN DECLARE SECTION END-EXEC.
369  *
370  01 WS-BYPASS-STATES-ARRAY.
371      05 WS-BYPASS-STATES OCCURS 50 TIMES PIC X(02).

```

371	01	USERID	PIC X(01) VALUE "/".
372	01	DBS	PIC X(04) VARYING.
373	01	WS-DBS-LEN	PIC X(01).
374	01	COMPILED-DATE	PIC X(25) VALUE SPACES.
375	01	WS-UETB-DP-FIELD-NAME	PIC X(60).
376	01	WS-UETB-DP-PROG-NAME	PIC X(30) VALUE "WCLOSS_STATS".
377	01	WS-CLPT-CL-TAX-ID-NO	PIC X(09).
378	01	WS-CLPT-CL-TAX-ID-TYPE	PIC X.
379	01	WS-CLPT-PTTP-CL-TYPE-CODE	PIC X(04).
380	01	WS-RVHS-CL-NO	PIC S9(07) COMP-3.
381	01	WS-RVHS-RSRV-CL-NO	PIC S9(07) COMP-3.
382	01	WS-RVHS-GB-ACCT	PIC S9(07) COMP-3.
383	01	WS-RVHS-POL-NO	PIC X(06).
384	01	WS-RVHS-POL-HOLD	PIC X(06).
385	01	WS-RVHS-CLASS-CODE	PIC X(06).
386	01	WS-RVHS-GB-STATE	PIC X(02).
387	01	WS-RVHS-GB-STATE-NO	PIC X(02).
388	01	WS-RVHS-CL-OCC-DATE	PIC X(11).
389	01	WS-RVHS-CL-OCCURRENCE-YYYY	PIC X(10).
390	01	WS-RVHS-CL-CLAIM-NO-TRIM	PIC X(08).
391	01	WS-RVHS-CL-CLAIM-NO	PIC X(08).
392	01	WS-RVHS-RSTP-CL-TYPE-CODE	PIC X(03).
393	01	WS-RVHS-CLIC-CL-CODE	PIC S9(02) COMP-3.
394	01	WS-RVHS-CLOSE-DATE	PIC X(08).
395	01	WS-RVHS-ORIG-CLOSE-DATE	PIC X(09).
396	01	WS-RVHS-REOPEN-DATE	PIC X(08).
397	01	WS-RVHS-WCCC-TYPE-CODE-C	PIC X(02).
398	01	WS-RVHS-WCCC-TYPE-CODE-N	PIC S9(02) COMP-3.
399	01	WS-RVHS-GB-POLY-EFF-DATE	PIC X(11).
400	01	WS-RVHS-GB-POLY-EXP-DATE	PIC X(11).
401	01	WS-RVHS-BSOR-GB-FM-ACCT	PIC S9(07) COMP-3.
402	01	WS-HOLD-BSOR-GB-FM-ACCT	PIC S9(07) COMP-3.
403	01	WS-HOLD-RVHS-RSRV-CL-NO	PIC S9(07) COMP-3.
404	01	WS-RVHS-CLOC-CL-NO	PIC S9(07) COMP-3.
405	01	WS-CLOSE-IND	PIC X.
406	01	WS-MCTB-GB-CODE-TYPE	PIC X(08) VALUE SPACES.
407	01	WS-CLOC-CL-JOINT-COV-PERCENT	PIC S9(03) COMP-3.
408	01	WS-CLPT-CL-PERM-DISABL-RATING	PIC S9(03) COMP-3.
409	01	WS-RSRV-CL-REHAB	PIC S9(09) COMP-3.
410	01	WS-CLOC-CL-S-CLAIM	PIC S9(09) COMP-3.
411	01	WS-SUBRO	PIC S9(07) COMP-3.
412	01	WS-SUBRO-PART	PIC S9(07) COMP-3.
413	01	WS-S-CLAIM	PIC S9(07) COMP-3.
414	01	WS-JOINT	PIC S9(07) COMP-3.
415	01	WS-UNLS-CL-PAID-VOC-REHAB	PIC S9(07) COMP-3.
416	01	WS-LAWSUIT-IND	PIC X(01) VALUE 'N'.
417	*		
418	01	WS_CLPT_CL_HIRE_DATE	PIC X(11).
419	01	WS-UNPL-XMIT-IND	PIC X(01).
420	01	WS-RVHS-ACT-CODE	PIC X(02) VALUE SPACES.
421	01	WS-RVHS-LOSS-CODE	PIC X(02) VALUE SPACES.
422	01	WS-RVHS-RECV-CODE	PIC X(02) VALUE SPACES.
423	01	WS-RVHS-COVG-CODE	PIC X(02) VALUE SPACES.
424	01	WS-RVHS-SETT-CODE	PIC X(02) VALUE SPACES.
425	01	WS-RVHS-FRAUD-CODE	PIC X(02) VALUE SPACES.
426	01	WS-SOURCE	PIC X(16) VALUE SPACES.

427		
428	01 WS-CAT-IND	PIC S9(01) VALUE ZERO.
429	01 WS-CAT-NUMBER	PIC S9(02) VALUE ZEROES.
430	01 WS-RLOC-NO	PIC S9(04).
431	01 WS-RLOC-HD	PIC S9(04).
432		
433	*	
434	*-----*	
435	* UNLS BIND VARIABLES *	
436	*-----*	
437	01 WS-UNLS-INSERTS	PIC S9(03) COMP-3.
438	01 WS-UNLS-CL-NO	PIC S9(07) COMP-3.
439	01 WS-UNLS-UNPL-WC-NO	PIC S9(07) COMP-3.
440	01 WS-UNLS-REPT-DATE	PIC X(11).
441	01 WS-UNLS-PREV-REP-IND	PIC S9(01) COMP-3.
442	01 WS-UNLS-COVG-CODE	PIC S9(02) COMP-3.
443	01 WS-UNLS-NUM-OF-CLMS	PIC S9(04) COMP-3.
444	01 WS-UNLS-CLM-ACDNT-DTE	PIC X(08).
445	01 WS-UNLS-CLM-NUMBER	PIC X(12).
446	01 WS-UNLS-CLM-STATUS	PIC X(01).
447	01 WS-UNLS-CLM-INJURY-CODE	PIC S9(02) COMP-3.
448	01 WS-UNLS-CLM-CAT-NO	PIC S9(02) VALUE 0 COMP-3.
449	01 WS-UNLS-CLM-SSN	PIC X(09).
450	01 WS-UNLS-CLM-HMO-TYPE	PIC X(02).
451	01 WS-UNLS-CLM-BODY-PART-CDE	PIC X(04).
452	01 WS-UNLS-CLM-NAT-OF-INJ-CDE	PIC X(04).
453	01 WS-UNLS-CLM-CSE-OF-INJ-CDE	PIC X(04).
454	01 WS-UNLS-CLM-OCCUP-DESC	PIC X(18).
455	01 WS-UNLS-CLM-VOC-REHAB-IND	PIC X(01).
456	01 WS-UNLS-CLM-LUMP-SUM-IND	PIC X(01).
457	01 WS-UNLS-CLM-FRAUD-IND	PIC X(02).
458	01 WS-UNLS-CLM-DEDUCT-CODE	PIC X(02).
459	01 WS-UNLS-CLM-IDMNTY-PYMT-AMT	PIC S9(09) COMP-3.
460	01 WS-UNLS-CLM-MED-PYMT-AMT	PIC S9(09) COMP-3.
461	01 WS-UNLS-CLM-CLAIMAINT-ATT-FEE	PIC S9(09) COMP-3.
462	01 WS-UNLS-CLM-EMPLOYER-ATT-FEE	PIC S9(09) COMP-3.
463	01 WS-UNLS-CLM-ALLOC-LAE-PD	PIC S9(09) COMP-3.
464	01 WS-UNLS-CLM-ALLOC-LAE-IN	PIC S9(09) COMP-3.
465	01 WS-UNLS-UPD-TYPE	PIC X(01).
466	01 WS-UNLS-LOSS-ACT	PIC X(02).
467	01 WS-UNLS-LOSS-TYPE	PIC X(02).
468	01 WS-UNLS-LOSS-RECOV	PIC X(02).
469	01 WS-UNLS-LOSS-COVG	PIC X(02).
470	01 WS-UNLS-LOSS-SETL	PIC X(02).
471	01 WS-UNLS-INDEMNITY-AMT	PIC S9(09) COMP-3.
472	01 WS-UNLS-MEDICAL-AMT	PIC S9(09) COMP-3.
473	01 WS-UNLS-CL-INJ-DESC-CD-CLS	PIC S9(02) COMP-3.
474	01 WS-UNLS-CL-AVG-WKLY-WAGE	PIC S9(05) COMP-3 VALUE 0.
475	01 WS-UNLS-CL-INCUR-VOC-REHAB	PIC S9(07) COMP-3.
476	01 WS-UNLS-CL-CA-GROSS-INCUR	PIC S9(09) COMP-3.
477	01 WS-UNLS-CL-PERM-DISABL-RATING	PIC S9(03) COMP-3.
478	*	
479	01 WS-UNLS-REPT-NO	PIC 9(02) VALUE 00.
480	01 WS-UNLS-REPT-NO-TECH	PIC 9(02) VALUE 00.
481	01 WS-UNLS-REPT-NO-UCLA	PIC 9(02) VALUE 00.
482	01 WS-UNLS-REPT-NO-RCVRY	PIC 9(02) VALUE 00.

483	01 WS-UNLS-CLM-NUMBER-HOLD	PIC X(12).
484	01 WS-PRVSLY-RPTD	PIC X(01).
485	01 WS-STATUS-OVERRIDE	PIC X(01).
486	*-----*	
487	*-----*	
488	*-----*	
489	01 WS-NCCI-VARIABLES.	
490	05 WS-INFC-GB-NO	PIC S9(07) COMP-3.
491	05 WS-RVHS-PLSM-GB-NO	PIC S9(07) COMP-3.
492	05 WS-RVHS-GB-CLAIMANT	PIC X(60).
493	05 WS-RVHS-GB-POLY-EFF-MMDDYYYY	PIC X(08).
494	05 WS-RVHS-GB-POLY-EFF-YYYYMMDD	PIC X(08).
495	05 WS-AGENT-ACCT	PIC S9(07) COMP-3.
496	05 WS-BSOR-GB-NAME	PIC X(60).
497	05 WS-NCCI-POLICY-NUMBER	PIC X(16).
498	05 WS-REWRITE	PIC X(02).
499	05 WS-REWRITE-COUNT	PIC S9(03) COMP-3.
500	05 WS-EXIT-GB-STD-PREM	PIC S9(10) COMP-3.
501	05 WS-INFC-GB-ANNUAL-PX	PIC S9(10) COMP-3.
502	05 WS-SYSDATE-MMDDYY	PIC X(06).
503	05 WS-SYSDATE-MMDDYYYY	PIC X(08).
504	05 WS-SYSDATE-YYDDD	PIC X(05).
505	05 WS-SYSDATE-TIME	PIC X(18).
506	05 WS-ASSM-GB-ASSMNT-CODE	PIC X(06).
507	05 WS-ASSM-GB-AMOUNT	PIC S9(10) COMP-3.
508	05 WS-INFC-WC-LIAB-LIMIT	PIC S9(11) COMP-3.
509	05 WS-BURT-WC-RATE-ARRAY OCCURS 18 TIMES	PIC S9(07) COMP-3.
510	05 WS-INFC-HSTP-GB-CODE-OF	PIC X(02).
511	05 WS-POLY-GB-PREV-POLNO	PIC X(15).
512	05 WS-EXIT-GB-WRIT-PREM	PIC S9(11) COMP-3.
513	05 WS-PLSM-GB-POL-PREFIX	PIC X(05).
514	05 WS-RVHS-GB-POL-PREFIX	PIC X(05).
515	05 WS-EXIT-WC-CLASS-ARRAY OCCURS 18 TIMES	PIC X(06).
516	05 WS-EXIT-GB-COV-AMOUNT OCCURS 18 TIMES	PIC S9(13) COMP-3.
517	05 WS-EXIT-GB-STD-ARRAY OCCURS 18 TIMES	PIC S9(11) COMP-3.
518	05 WS-EXIT-WC-EXP-MOD	PIC S9(02)V9(04) COMP-3.
519	05 WS-RVHS-WC-WEEKLY-WAGE	PIC S9(07) COMP-3.
520	05 WS-RVHS-POBI-CL-TYPE-CODE	PIC X(04).
521	05 WS-RVHS-CL-OCCURRENCE-MMDDYYYY	PIC X(08).
522	05 WS-RVHS-COLT-CL-TYPE-CODE	PIC X(04).
523	05 WS-RVHS-NOSI-CL-TYPE-CODE	PIC X(04).
524	05 WS-EXIT-WC-CLASS-CODE	PIC X(05).
525	05 WS-RVHS-CL-REPORT-MMDDYYYY	PIC X(08).
526	05 WS-LSRV-CL-RESV-AMOUNT	PIC S9(07)V99 COMP-3.
527	05 WS-LSPD-CL-PAID-AMOUNT	PIC S9(07)V99 COMP-3.
528	05 WS-LSPD-EXP-PAYMENT-AMT	PIC S9(07)V99 COMP-3.
529	05 WS-EXPENSE-RESERVE	PIC S9(07)V99 COMP-3.
530	05 WS-LOCT-PC-LOCATION-CODE	PIC S9(03) COMP-3.
531	*-----*	
532	01 WS-RVHS-GB-CLASS-CODE-ASWG	PIC X(05).
533	01 WS-LSPD-CL-PAID-AMOUNT-ASWG	PIC S9(07)V99 COMP-3.
534	01 WS-LSPD-DRTP-GB-TYPE-ASWG	PIC X(04).
535	01 WS-LSPD-DFCM-CL-TYPE-CODE-ASWG	PIC X(03).
536	01 WS-RSRV-CLDC-CL-CODE-ASWG	PIC X(03).
537	01 WS-COV-B-CHECK	PIC X.
538	01 WS-COV-A-CHECK	PIC X.

```

539 01 WS-SUBRO-CHECK PIC X.
540 01 WS-SIF-CHECK PIC X.
541 *
542 01 WS-CURRENT-RESV PIC S9(11)V99 COMP-3.
543 01 WS-PAID PIC S9(11)V99 COMP-3.
544 *
545 *BEGIN PLI31530.3248***
546 01 WS-WC-POL PIC S9(3).
547 01 WS-DEDUCT-IND PIC X(01).
548 01 WS-POLICY-HOLD PIC X(06).
549 01 WS-MESSAGE PIC X(255).
550 01 WS-MEANING PIC X(05).
551 01 WS-GROSS-PAID PIC S9(11)V99 COMP-3.
552 01 WS-DEDUCT PIC S9(11)V99 COMP-3.
553 01 WS-UNLS-CL-DEDUCT-AMT PIC S9(11)V99 COMP-3.
554
555 *END PLI31530.3248***
556 *
557 01 WS-RERUN-DATE PIC X(11).
558 01 WS-RPT-MONTH PIC X(03).
559 01 WS-BEG-DATE PIC X(11).
560 01 WS-BEG-DATE-YYYY PIC X(08).
561 01 WS-BEG-DATE-18 PIC X(11).
562 01 WS-BEG-DATE-12 PIC X(11).
563 01 WS-BEG-DATE-11 PIC X(11).
564 01 WS-END-DATE PIC X(11).
565 01 WS-END-DATE-YYYY PIC X(08).
566 01 WS-END-DATE-YY PIC X(11).
567 01 WS-END-DATE-18 PIC X(11).
568 01 WS-DTCL-GB-BEG-DATE PIC X(11).
569 01 WS-DTCL-GB-END-DATE PIC X(11).
570
571 *
572 01 WS-STATE-IN PIC X(12).
573 01 STATE1 PIC X(02).
574 01 STATE2 PIC X(02).
575 01 STATE3 PIC X(02).
576 01 STATE4 PIC X(02).
577 01 STATE5 PIC X(02).
578 01 STATE6 PIC X(02).
579 01 WS-UPDATE-SWITCH PIC X VALUE SPACES.
580 *
581 *-----*
582 *-----*
583 EXEC SQL END DECLARE SECTION END-EXEC.
584 *
585 EXEC SQL INCLUDE SQLCA5 END-EXEC.
586 *
587 *-----*
588 * HOLD DATA AREAS *
589 *-----*
590 01 REPORT-FILENAME PIC X(150).
591 01 REPORT-FILENAME1 PIC X(150).
592 01 REPORT-FILENAME2 PIC X(150).
593 01 CLAIM-REPORT-FILENAME PIC X(150).
594 01 HOLD-CLAIM-OUTPUT PIC X(303).

```

595	01	START-STRING	PIC S9(02) COMP.
596	01	STRING-SIZE	PIC S9(02) COMP.
597	01	WS-CREATE-DUMMY	PIC X.
598	01	HOLD-CLAIM-NO	PIC X(08).
599	01	LEFTOVER-STRING	PIC X(34) VALUE SPACES.
600	01	POLNUM-DATE-TIME-STRING	PIC X(34) VALUE SPACES.
601	01	WHAT-SUB	PIC S9(02) COMP.
602	01	HOLD-UNLS-PREV-REP-IND	PIC S9(1) COMP-3.
603	01	HOLD-UNLS-UPD-TYPE	PIC X(1).
604	01	HOLD-UNLS-UNPL-WC-NO	PIC S9(7) COMP-3.
605	01	HOLD-RVHS-CLASS-CODE	PIC X(6).
606	01	HOLD-UNLS-REPT-DATE	PIC X(11).
607	01	HOLD-UNLS-CLM-NUMBER	PIC X(12).
608	*		
609	01	WS-TID-S.	
610	05	WS-TID-S3	PIC XXX.
611	05	FILLER	PIC X VALUE '-'. PIC XX.
612	05	WS-TID-S2	
613	05	FILLER	PIC X VALUE '-'. PIC XXXX.
614	05	WS-TID-S4	
615	*		
616	01	WS-TID-F.	
617	05	WS-TID-F2	PIC XX.
618	05	FILLER	PIC X VALUE '-'. PIC X(07).
619	05	WS-TID-F7	
620	*		
621	01	WS-TID	PIC X(11).
622	*		
623	01	WS-SYS-DATE.	
624	05	WS-SYS-YY1	PIC XX.
625	05	WS-SYS-MM	PIC X(02).
626	05	WS-SYS-DD	PIC X(02).
627	*		
628	01	WS-HOLD-DATE.	
629	05	WS-HOLD-MM	PIC X(02).
630	05	WS-HOLD-SLASH1	PIC X(01) VALUE "/".
631	05	WS-HOLD-DD	PIC X(02).
632	05	WS-HOLD-SLASH2	PIC X(01) VALUE "/".
633	05	WS-HOLD-YY1	PIC X(02).
634	*		
635	01	WS-SYS-TIME.	
636	05	WS-SYS-HR	PIC X(02).
637	05	WS-SYS-MN	PIC X(02).
638	*		
639	01	WS-HOLD-TIME.	
640	05	WS-HOLD-HR	PIC X(02).
641	05	WS-HOLD-COLON	PIC X(01) VALUE ":".
642	05	WS-HOLD-MN	PIC X(02).
643	*		
644	01	WS-HOLD-STATE	PIC X(02).
645	01	WS-DATE-OPT	PIC X(01).
646	01	WS-ERR-MSG	PIC X(48).
647	01	WS-PAGE	PIC 9(04) COMP-3 VALUE 0.
648	01	WS-LINE-COUNT	PIC 9(02) COMP-3.
649	01	WS-LINE-COUNT2	PIC 99 COMP-3.
650	01	WS-PAGE-COUNT	PIC 99 COMP-3.

```

651      *
652      01 WS-TOT-CUR-RESV          PIC S9(11)V99 COMP-3.
653      01 WS-TOT-PAID             PIC S9(11)V99 COMP-3.
654      01 WS-TOT-DEDUCT           PIC S9(11)V99 COMP-3.
655      01 WS-ALL-CUR-RESV        PIC S9(11)V99 COMP-3.
656      01 WS-ALL-PAID            PIC S9(11)V99 COMP-3.
657      01 WS-ALL-DEDUCT          PIC S9(11)V99 COMP-3.
658      01 WS-MED-COUNT           PIC 9(06).
659      01 WS-MED-TOTAL           PIC 9(06).
660      01 WS-INDEM-COUNT          PIC 9(06).
661      01 WS-INDEM-TOTAL          PIC 9(06).
662      01 WS-TOTAL-CLAIMS         PIC 9(06).
663      01 WS-ALL-CLAIMS          PIC 9(06).
664      01 WS-RESV-FLAG           PIC X(01).
665      01 WS-RPT-SEQ             pic x(01).
666
667
668      *
669      *-----*
670      *          NORMAL WORK AREAS          *
671      *-----*
672      *
673      *-----*
674      *          SWITCHES                   *
675      *-----*
676      *
677      01 SW-DONE                  PIC X(01) VALUE "N".
678      88 DONE                     VALUE "Y".
679      *
680      01 SW-ABORT                 PIC X(01) VALUE "N".
681      88 ABORT                    VALUE "Y".
682      *
683      01 SW-EOF-CURS              PIC X(01) VALUE "N".
684      88 EOF-CURS                 VALUE "Y".
685      *
686      01 LOSS-EOF-SW              PIC X(01) VALUE "N".
687      88 LOSS-EOF                 VALUE "Y".
688      *
689      01 SW-PROCESS-STATE-RPT2    PIC X(01) VALUE "N".
690      88 PROCESS-STATE-RPT2       VALUE "Y".
691      *
692      01 SW-PROCESS-NEXT-STATE    PIC X(01) VALUE "N".
693      88 PROCESS-NEXT-STATE       VALUE "Y".
694      *
695      01 WS-DATA-OPTION            PIC X(01).
696      88 CREATE-DATA              VALUE 'Y'.
697      88 NO-DATA                  VALUE 'N'.
698      *
699      01 WS-RERUN                 PIC X.
700      *
701      01 WS-BYPASS-STATUS          PIC S9(01) COMP VALUE 0.
702      88 BYPASS-DATA              VALUE +1.
703      *
704      *-----*
705      *          PRINT LINES              *
706      *-----*

```

```

707 *
708 01 LINE-DETAIL.
709     05 FILLER                PIC X(05) VALUE SPACES.
710     05 P-POL-NO              PIC X(06).
711     05 FILLER                PIC X(03) VALUE SPACES.
712     05 P-EFF-DATE            PIC X(13).
713     05 P-CLASS-CODE          PIC X(05).
714     05 FILLER                PIC X(01) VALUE SPACES.
715     05 P-OCC-DATE            PIC X(11).
716     05 FILLER                PIC X(03) VALUE SPACES.
717     05 P-CLAIM-NO            PIC X(08).
718     05 FILLER                PIC X(03) VALUE SPACES.
719     05 P-RESV-TYPE            PIC X(03).
720     05 FILLER                PIC X(03) VALUE SPACES.
721     05 P-INJURY              PIC X(03).
722     05 FILLER                PIC X(04) VALUE SPACES.
723     05 P-STATUS              PIC X(03).
724     05 FILLER                PIC X(03) VALUE SPACES.
725     05 P-CURR-RESV           PIC Z,ZZZ,Z99.99-.
726     05 FILLER                PIC X(03) VALUE SPACES.
727     05 P-PAID-AMT            PIC Z,ZZZ,Z99.99-.
728     05 FILLER                PIC X(04) VALUE SPACES.
729     05 P-DED-IND             PIC X(01) VALUE SPACES.
730     05 FILLER                PIC X(02) VALUE SPACES.
731     05 P-DEDUCT-AMT          PIC Z,ZZZ,Z99.99-.
732     05 FILLER                PIC X(01) VALUE SPACES.
733     05 P-DESCRIPT            PIC X(16) VALUE SPACES.
734     05 FILLER                PIC X(10) VALUE SPACES.
735
736 *
737 01 E-HEAD-1.
738     05 FILLER                PIC X(30)
739     VALUE "----- ".
740     05 FILLER                PIC X(30)
741     VALUE "----- ".
742     05 FILLER                PIC X(18)
743     VALUE "EXCEPTION REPORT ".
744     05 FILLER                PIC X(52)
745     VALUE "----- ".
746     05 FILLER                PIC X(6)
747     VALUE 'PAGE '.
748     05 E-PAGE                PIC ZZ9
749     VALUE ZERO.
750
751 *
752 01 E-HEAD-2.
753     05 FILLER                PIC X(30) VALUE SPACES.
754     05 FILLER                PIC X(40) VALUE
755     "THE CLAIMS LISTED BELOW HAVE ERRORS.".
756     *****-- "THE CLAIMS LISTED BELOW HAVE NO MATCHING".
757     05 FILLER                PIC X(100) VALUE
758     "CHECK THE POLICY TABLE AND PAID AMOUNTS.".
759     *****-- " RECORD IN THE UNIT_STATS_POLICY TABLE".
760
761 *
762 01 EXCEPTION-DETAIL.
763     05 FILLER                PIC X(05) VALUE SPACES.
764     05 E-POL-NO              PIC X(06).

```


763	05 FILLER	PIC X(03) VALUE SPACES.
764	05 E-EFF-DATE	PIC X(13).
765	05 E-CLASS-CODE	PIC X(05).
766	05 FILLER	PIC X(05) VALUE SPACES.
767	05 E-TYPE-CODE	PIC X(02).
768	05 FILLER	PIC X(04) VALUE SPACES.
769	05 E-STATE	PIC X(03).
770	05 FILLER	PIC X(03) VALUE SPACES.
771	05 E-OCC-DATE	PIC X(11).
772	05 FILLER	PIC X(03) VALUE SPACES.
773	05 E-CLAIM-NO	PIC X(08).
774	05 FILLER	PIC X(03) VALUE SPACES.
775	05 E-RESV-TYPE	PIC X(03).
776	05 FILLER	PIC X(03) VALUE SPACES.
777	05 E-INJURY	PIC X(03).
778	05 FILLER	PIC X(06) VALUE SPACES.
779	05 E-STATUS	PIC X(03) VALUE SPACES.
780	05 FILLER	PIC X(03) VALUE SPACES.
781	05 E-CURR-RESV	PIC Z,ZZZ,Z99.99-.
782	05 FILLER	PIC X(03) VALUE SPACES.
783	05 E-PAID-AMT	PIC Z,ZZZ,Z99.99-.
784	05 FILLER	PIC X(13) VALUE SPACES.
785	*	
786	01 P-HEAD-1.	
787	05 FILLER	PIC X(13) VALUE "WCLOSS_STATS".
788	05 FILLER	PIC X(25) VALUE SPACES.
789	05 P-COMPANY-NAME	PIC X(60).
790	05 FILLER	PIC X(25) VALUE SPACES.
791	05 FILLER	PIC X(05) VALUE "PAGE".
792	05 P-PGCOUNT	PIC ZZZ9.
793		
794	01 P-HEAD-ASEQ.	
795	05 FILLER	PIC X(27) VALUE
796		"WCLOSS_STATS (ASEQUENTIAL)".
797	05 FILLER	PIC X(11) VALUE SPACES.
798	05 P-COMPANY-NAME-ASEQ	PIC X(60).
799	05 FILLER	PIC X(25) VALUE SPACES.
800	05 FILLER	PIC X(05) VALUE "PAGE".
801	05 P-PGCOUNT-ASEQ	PIC ZZZ9.
802	*	
803	01 P-HEAD-2.	
804	05 FILLER	PIC X(05) VALUE "DATE".
805	05 P-RUN-DATE	PIC X(08).
806	05 FILLER	PIC X(45) VALUE SPACES.
807	05 P-SUB-HEAD-1	PIC X(20) VALUE
808		"WORK COMP UNIT STATS".
809	05 FILLER	PIC X(44) VALUE SPACES.
810	05 FILLER	PIC X(05) VALUE "TIME".
811	05 P-TIME	PIC X(05).
812	*	
813	01 P-HEAD-3.	
814	05 FILLER	PIC X(52) VALUE SPACES.
815	05 FILLER	PIC X(20) VALUE
816		" FOR THE STATE OF : ".
817	05 P-STATE1	PIC X(02).
818	05 FILLER	PIC X(60) VALUE SPACES.

```

819 *
820 01 P-HEAD-4.
821     05 FILLER          PIC X(05) VALUE SPACES.
822     05 FILLER          PIC X(09) VALUE "POLICY".
823     05 FILLER          PIC X(13) VALUE "POLICY".
824     05 FILLER          PIC X(05) VALUE "CLASS".
825     05 FILLER          PIC X(11) VALUE " DATE OF ".
826     05 FILLER          PIC X(03) VALUE SPACES.
827     05 FILLER          PIC X(09) VALUE " CLAIM NO".
828     05 FILLER          PIC X(02) VALUE SPACES.
829     05 FILLER          PIC X(04) VALUE "RESV".
830     05 FILLER          PIC X(01) VALUE SPACES.
831     05 FILLER          PIC X(06) VALUE "INJURY".
832     05 FILLER          PIC X(01) VALUE SPACES.
833     05 FILLER          PIC X(06) VALUE "STATUS".
834     05 FILLER          PIC X(01) VALUE SPACES.
835     05 FILLER          PIC X(12) VALUE "GROSS ".
836     05 FILLER          PIC X(03) VALUE SPACES.
837     05 FILLER          PIC X(16) VALUE " GROSS PD. AMT ".
838     05 FILLER          PIC X(01) VALUE SPACES.
839     05 FILLER          PIC X(04) VALUE "DED?".
840     05 FILLER          PIC X(01) VALUE SPACES.
841     05 FILLER          PIC X(13) VALUE "DEDUCTIBLE ".
842     05 FILLER          PIC X(06) VALUE "DESC. ".
843 *
844 01 P-HEAD-5.
845     05 FILLER          PIC X(05) VALUE SPACES.
846     05 FILLER          PIC X(09) VALUE "NUMBER ".
847     05 FILLER          PIC X(13) VALUE "EFFECTIVE".
848     05 FILLER          PIC X(05) VALUE "CODE ".
849     05 FILLER          PIC X(01) VALUE SPACES.
850     05 FILLER          PIC X(10) VALUE "OCCURRENCE".
851     05 FILLER          PIC X(11) VALUE SPACES.
852     05 FILLER          PIC X(03) VALUE SPACES.
853     05 FILLER          PIC X(04) VALUE "TYPE".
854     05 FILLER          PIC X(01) VALUE SPACES.
855     05 FILLER          PIC X(05) VALUE "CODE ".
856     05 FILLER          PIC X(09) VALUE SPACES.
857     05 FILLER          PIC X(10) VALUE "RESERVE ".
858     05 FILLER          PIC X(27) VALUE SPACES.
859     05 FILLER          PIC X(11) VALUE "AMOUNT".
860 *
861 01 P-HEAD-6.
862     05 FILLER          PIC X(05) VALUE SPACES.
863     05 FILLER          PIC X(09) VALUE "-----".
864     05 FILLER          PIC X(13) VALUE "-----".
865     05 FILLER          PIC X(06) VALUE "-----".
866     05 FILLER          PIC X(13) VALUE "-----".
867     05 FILLER          PIC X(09) VALUE
868 "-----".
869     05 FILLER          PIC X(02) VALUE SPACES.
870     05 FILLER          PIC X(04) VALUE
871 "----".
872     05 FILLER          PIC X(01) VALUE SPACES.
873     05 FILLER          PIC X(06) VALUE
874 "-----".

```

```

875      05 FILLER          PIC X(01) VALUE SPACES.
876      05 FILLER          PIC X(06) VALUE
877      "-----".
878      05 FILLER          PIC X(01) VALUE SPACES.
879      05 FILLER          PIC X(13) VALUE
880      "-----".
881      05 FILLER          PIC X(03) VALUE SPACES.
882      05 FILLER          PIC X(13) VALUE
883      "-----".
884      05 FILLER          PIC X(03) VALUE SPACES.
885      05 FILLER          PIC X(04) VALUE "----".
886      05 FILLER          PIC X(01) VALUE SPACES.
887      05 FILLER          PIC X(13) VALUE "-----".
888      05 FILLER          PIC X(06) VALUE "----".
889
890      *
891      01 P-HEAD-7.
892          05 FILLER          PIC X(02) VALUE SPACES.
893          05 FILLER          PIC X(08) VALUE
894          "TOTALS: ".
895          05 FILLER          PIC X(65) VALUE SPACES.
896          05 P-TOTAL-RESERVE PIC ZZZ,ZZZ,ZZ9.99-.
897          05 FILLER          PIC X(01) VALUE SPACES.
898          05 P-TOTAL-PAID    PIC ZZZ,ZZZ,ZZ9.99-.
899          05 FILLER          PIC X(05) VALUE SPACES.
900          05 P-TOTAL-DEDUCT  PIC ZZZ,ZZZ,ZZ9.99-.
901          05 FILLER          PIC X(01) VALUE SPACES.
902      *
903      01 P-HEAD-8.
904          05 FILLER          PIC X(27) VALUE
905          " TOTAL MEDICAL CLAIMS : ".
906          05 P-TOTAL-MED     PIC ZZZZZZZZ9.
907          05 FILLER          PIC X(90) VALUE SPACES.
908      *
909      01 P-HEAD-9.
910          05 FILLER          PIC X(27) VALUE
911          " TOTAL INDEMNITY CLAIMS : ".
912          05 P-TOTAL-INDEM    PIC ZZZZZZZZ9.
913          05 FILLER          PIC X(90) VALUE SPACES.
914      *
915      01 P-HEAD-10.
916          05 FILLER          PIC X(27) VALUE
917          " TOTAL CLAIMS      : ".
918          05 P-TOTAL-CLAIMS   PIC ZZZZZZZZ9.
919          05 FILLER          PIC X(90) VALUE SPACES.
920      *
921      01 P-HEAD-11.
922          05 FILLER          PIC X(45) VALUE SPACES.
923          05 FILLER          PIC X(42) VALUE
924          "***** END OF REPORT *****".
925          05 FILLER          PIC X(45) VALUE SPACES.
926      *
927      01 P-HEAD-12.
928          05 FILLER          PIC X(30) VALUE SPACES.
929          05 FILLER          PIC X(24) VALUE
930          "*****".

```

```

931      05 FILLER          PIC X(25) VALUE
932      " TOTALS FOR ALL CLAIMS ".
933      05 FILLER          PIC X(24) VALUE
934      "*****"
935      05 FILLER          PIC X(29) VALUE SPACES.
936
937
938      01 X-HEAD-4.
939      05 FILLER          PIC X(05) VALUE SPACES.
940      05 FILLER          PIC X(09) VALUE "POLICY".
941      05 FILLER          PIC X(13) VALUE "POLICY".
942      05 FILLER          PIC X(10) VALUE "CLASS".
943      05 FILLER          PIC X(06) VALUE "LOSS".
944      05 FILLER          PIC X(06) VALUE "STATE".
945      05 FILLER          PIC X(10) VALUE " DATE OF ".
946      05 FILLER          PIC X(03) VALUE SPACES.
947      05 FILLER          PIC X(08) VALUE "CLAIM NO".
948      05 FILLER          PIC X(03) VALUE SPACES.
949      05 FILLER          PIC X(04) VALUE "RESV".
950      05 FILLER          PIC X(03) VALUE SPACES.
951      05 FILLER          PIC X(06) VALUE "INJURY".
952      05 FILLER          PIC X(03) VALUE SPACES.
953      05 FILLER          PIC X(06) VALUE "STATUS".
954      05 FILLER          PIC X(03) VALUE SPACES.
955      05 FILLER          PIC X(13) VALUE " CURRENT ".
956      05 FILLER          PIC X(03) VALUE SPACES.
957      05 FILLER          PIC X(13) VALUE " PAID AMT ".
958      05 FILLER          PIC X(19) VALUE " FED-ID/SSN ".
959      *
960      01 X-HEAD-5.
961      05 FILLER          PIC X(04) VALUE "XMIT".
962      05 FILLER          PIC X(01) VALUE SPACES.
963      05 FILLER          PIC X(09) VALUE "NUMBER ".
964      05 FILLER          PIC X(13) VALUE "EFFECTIVE".
965      05 FILLER          PIC X(10) VALUE "CODE ".
966      05 FILLER          PIC X(06) VALUE "COVG. ".
967      05 FILLER          PIC X(06) VALUE SPACES.
968      05 FILLER          PIC X(10) VALUE "OCCURRENCE".
969      05 FILLER          PIC X(11) VALUE SPACES.
970      05 FILLER          PIC X(03) VALUE SPACES.
971      05 FILLER          PIC X(04) VALUE "TYPE".
972      05 FILLER          PIC X(03) VALUE SPACES.
973      05 FILLER          PIC X(06) VALUE " CODE ".
974      05 FILLER          PIC X(12) VALUE SPACES.
975      05 FILLER          PIC X(13) VALUE " RESERVE ".
976      05 FILLER          PIC X(19) VALUE SPACES.
977      *
978      *
979      01 MAX          PIC S9(03) COMP.
980      01 COUNTER      PIC S9(03) COMP.
981      * COPY "SYS$LOGIN:DISP_NAME.COB".
982      *-----*
983      *          0000-MAINLINE          *
984      *-----*
985      PROCEDURE DIVISION.
986      *

```

```

987 0000-MAINLINE.
988 *
989 EXEC SQL DECLARE ORA1 DATABASE END-EXEC.
990
991 * DISPLAY DISP-NAME.
992 PERFORM 0100-INIT THRU 0100-EXIT.
993 PERFORM 1000-PROCESS-REPORT THRU 1000-EXIT UNTIL EOF-CURS.
994 PERFORM 2500-STATE-BREAK THRU 2500-EXIT.
995 *
996 PERFORM 5000-END-REPORT-TOTALS THRU 5000-EXIT.
997 *
998 CLOSE REPORT-OUT,
999 REPORT-OUT2,
1000 REPORT-OUT3.
1001 * IF CREATE-DATA CLOSE CLAIM.
1002 *
1003 DISPLAY "ABENDSYM:NO".
1004 STOP RUN.
1005 *
1006 0000-EXIT.
1007 EXIT.
1008 *-----|0100-INIT|-----*
1009 *-----*
1010 * LOG PROGRAM ON ORACLE. *
1011 *-----*
1012 *
1013 0100-INIT.
1014
1015 DISPLAY 1 UPON AUG-NUM.
1016 ACCEPT DBS-ARR FROM AUG-VAL.
1017 DISPLAY 2 UPON AUG-NUM.
1018 ACCEPT WS-DBS-LEN FROM AUG-VAL.
1019 *
1020 MOVE WS-DBS-LEN TO DBS-LEN.
1021 *
1022 EXEC SQL AT ORA1
1023 CONNECT :USERID
1024 AT ORA1 USING :DBS
1025 END-EXEC.
1026 *
1027 IF SQLCODE NOT EQUAL 0
1028 MOVE "*** ERROR ** TRYING TO LOG ON TO ORACLE 0100" TO
1029 WS-ERR-MSG
1030 PERFORM 9900-ABORT THRU 9900-EXIT.
1031 *
1032 *-----*
1033 * RETRIEVE LOGICALS *
1034 *-----*
1035
1036 MOVE "OUTFILE" TO WS-UETB-DP-FIELD-NAME.
1037 CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME
1038 REPORT-FILENAME
1039 WS-UETB-DP-PROG-NAME.
1040
1041 MOVE "OUTFILE1" TO WS-UETB-DP-FIELD-NAME.
1042 CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME

```

```

1043             REPORT-FILENAME1
1044             WS-UETB-DP-PROG-NAME.
1045
1046             MOVE "OUTFILE2"      TO WS-UETB-DP-FIELD-NAME.
1047             CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME
1048             REPORT-FILENAME2
1049             WS-UETB-DP-PROG-NAME.
1050
1051             DISPLAY 'OUTFILE --> ' REPORT-FILENAME.
1052             DISPLAY 'OUTFILE1 --> ' REPORT-FILENAME1.
1053             DISPLAY 'OUTFILE2 --> ' REPORT-FILENAME2.
1054
1055
1056             OPEN OUTPUT REPORT-OUT,
1057             REPORT-OUT2,
1058             REPORT-OUT3.
1059 *
1060             MOVE FUNCTION WHEN-COMPILED TO COMPILED-DATE.
1061 *
1062             EXEC SQL AT ORA1
1063             SELECT TO_CHAR(TO_DATE(SUBSTR(:COMPILED-DATE,1,12)
1064             -           , 'YYYYMMDDHH24MI'), 'DD-MON-YYYY HH:MI')
1065             INTO :COMPILED-DATE
1066             FROM DUAL
1067             END-EXEC.
1068 *
1069             DISPLAY COMPILED-DATE.
1070 *
1071             IF SQLCODE NOT = 0 THEN
1072             DISPLAY 'ERROR IN COMPILED DATE DISPLAY'
1073             PERFORM 9900-ABORT THRU 9900-EXIT
1074             END-IF.
1075
1076             ACCEPT WS-SYS-DATE FROM DATE
1077 *
1078             MOVE WS-SYS-YY1 TO WS-HOLD-YY1.
1079             MOVE WS-SYS-MM TO WS-HOLD-MM.
1080             MOVE WS-SYS-DD TO WS-HOLD-DD.
1081 *
1082             MOVE WS-HOLD-DATE TO P-RUN-DATE.
1083 *
1084             ACCEPT WS-SYS-TIME FROM TIME.
1085             MOVE WS-SYS-HR TO WS-HOLD-HR.
1086             MOVE WS-SYS-MN TO WS-HOLD-MN.
1087             MOVE WS-HOLD-TIME TO P-TIME.
1088 *
1089             MOVE "DATEOPT" TO WS-UETB-DP-FIELD-NAME.
1090             CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME,
1091             WS-DATE-OPT,
1092             WS-UETB-DP-PROG-NAME.
1093
1094             DISPLAY 'DATEOPT --> '; WS-DATE-OPT.
1095 *
1096             MOVE "STATEIN" TO WS-UETB-DP-FIELD-NAME.
1097             CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME,
1098             WS-STATE-IN,

```



```

1099             WS-UETB-DP-PROG-NAME.
1100
1101     DISPLAY 'STATEIN --> '; WS-STATE-IN.
1102 *
1103     IF WS-STATE-IN = 'XX' THEN
1104         MOVE 'XX' TO STATE1
1105         MOVE 'XX' TO STATE2
1106         MOVE 'XX' TO STATE3
1107         MOVE 'XX' TO STATE4
1108         MOVE 'XX' TO STATE5
1109         MOVE 'XX' TO STATE6
1110     ELSE
1111         MOVE WS-STATE-IN (1:2) TO STATE1
1112         MOVE WS-STATE-IN (3:2) TO STATE2
1113         MOVE WS-STATE-IN (5:2) TO STATE3
1114         MOVE WS-STATE-IN (7:2) TO STATE4
1115         MOVE WS-STATE-IN (9:2) TO STATE5
1116         MOVE WS-STATE-IN (11:2) TO STATE6
1117     END-IF
1118 *
1119     IF WS-DATE-OPT = 'Y' THEN
1120
1121         MOVE "ENDDATE" TO WS-UETB-DP-FIELD-NAME.
1122         CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME,
1123             WS-END-DATE,
1124             WS-UETB-DP-PROG-NAME
1125         DISPLAY 'ENDDATE --> '; WS-END-DATE.
1126 *
1127     IF WS-DATE-OPT = 'Y' THEN
1128         PERFORM 0120-FORMAT-DATES THRU 0120-EXIT
1129     ELSE
1130         PERFORM 0110-GET-DATES THRU 0110-EXIT.
1131 *
1132     MOVE "DATAOPT" TO WS-UETB-DP-FIELD-NAME.
1133     CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME,
1134         WS-DATA-OPTION,
1135         WS-UETB-DP-PROG-NAME.
1136     DISPLAY 'DATAOPT --> '; WS-DATA-OPTION.
1137 *
1138     MOVE "RERUNOPT" TO WS-UETB-DP-FIELD-NAME.
1139     CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME,
1140         WS-RERUN,
1141         WS-UETB-DP-PROG-NAME.
1142     DISPLAY 'RERUNOPT --> '; WS-RERUN.
1143 *
1144     MOVE "RERUNDATE" TO WS-UETB-DP-FIELD-NAME.
1145     CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME,
1146         WS-RERUN-DATE,
1147         WS-UETB-DP-PROG-NAME.
1148     DISPLAY 'RERUNDATE --> '; WS-RERUN-DATE.
1149
1150
1151
1152
1153
1154

```

```

1155
1156
1157 *
1158 IF WS-RERUN = 'Y'
1159     NEXT SENTENCE
1160 ELSE
1161     MOVE '31-DEC-3000' TO WS-RERUN-DATE.
1162 *
1163     MOVE "UPDATESWITCH" TO WS-UETB-DP-FIELD-NAME.
1164     CALL "LIBSYS_TRNLOG" USING WS-UETB-DP-FIELD-NAME,
1165                               WS-UPDATE-SWITCH,
1166                               WS-UETB-DP-PROG-NAME.
1167     DISPLAY 'UPDATESWITCH --> '; WS-UPDATE-SWITCH.
1168 *
1169     IF WS-DATE-OPT = 'Y'
1170     THEN
1171         MOVE WS-END-DATE-YY TO WS-UNLS-REPT-DATE
1172     ELSE
1173         MOVE WS-DTCL-GB-END-DATE TO WS-UNLS-REPT-DATE
1174     END-IF.
1175 *
1176     IF SQLCODE NOT EQUAL 0 AND SQLCODE NOT = 1403
1177     THEN
1178         MOVE "***ERROR ** ERROR IN 0100"
1179         TO WS-ERR-MSG
1180         PERFORM 9900-ABORT THRU 9900-EXIT
1181     END-IF.
1182
1183 *-----*
1184 *          DECLARE CURSOR          *
1185 *-----*
1186 *
1187 * THIS CURSOR IS OPENED ONLY IF THE RERUN PARM IS NOT USED
1188 *
1189 * THE DECODE(SIGN... FUNCTION ON THE DATE COLUMNS HAS THE EFFECT OF
1190 * NULLING
1191 * THE DATES OUT IF THE RUN IS A RERUN AND THE DATES HIT AFTER
1192 * THE END DATE OF THE RERUN.
1193 *
1194 *          AND RVHS_GB_POL_NUMBER
1195 *          IN (SELECT UNPL_WC_POLY_NUMBER
1196 *              FROM UNPL
1197 *              WHERE UNPL_WC_REPORT_DATE
1198 *                  = TO_CHAR(TO_DATE(
1199 *                      :WS-END-DATE,'DD-MON-YYYY'),
1200 *                      'DD-MON-YY'))
1201 *
1202 *          AND RVHS_GB_POL_NUMBER = '500246'
1203 *
1204 EXEC SQL AT ORA1
1205 DECLARE CURS_WC_LOSS CURSOR FOR
1206     SELECT RVHS_CL_NO,
1207           RVHS_RSRV_CL_NO,
1208           RVHS_BSOR_GB_ACCOUNT,
1209           RVHS_GB_POL_NUMBER,
1210           RVHS_GB_POL_PREFIX,

```

```

1210         LPAD(RVHS_GB_CLASS_CODE,4,'0'),
1211         RVHS_GB_STATE,
1212         RVHS_CL_OCCURRENCE_DATE,
1213         TO_CHAR(TO_DATE(RVHS_CL_OCCURRENCE_DATE),
1214                 'YYYYMMDD'),
1215         TO_CHAR(TO_DATE(RVHS_CL_OCCURRENCE_DATE),
1216                 'MMDDYYYY'),
1217         RVHS_CL_CLAIMN_NO,
1218         LTRIM(LTRIM(RVHS_CL_CLAIMN_NO,'W'),'0'),
1219         RVHS_RSTP_CL_TYPE_CODE,
1220         DECODE (TO_NUMBER (SUBSTR (RVHS_CLIC_CL_CODE, 1, 1)), 0,
1221                 TO_NUMBER (SUBSTR (RVHS_CLIC_CL_CODE, 2, 1)),
1222                 TO_NUMBER (SUBSTR (RVHS_CLIC_CL_CODE, 1, 1))) ,
1223         TO_CHAR(TO_DATE(RVHS_CL_CLOSE_DATE),'YYYYMMDD'),
1224         TO_CHAR(TO_DATE(RVHS_CL_REOPEN_DATE),'YYYYMMDD'),
1225         RVHS_WCCC_CL_TYPE_CODE,
1226         RVHS_WCCC_CL_TYPE_CODE,
1227         RVHS_GB_POLY_EFF_DATE,
1228         RVHS_GB_POLY_EXP_DATE,
1229         RVHS_BSOR_GB_FM_ACCT,
1230         SUBSTR(RVHS_GB_CLAIMANT,1,60),
1231         TO_CHAR(TO_DATE(RVHS_GB_POLY_EFF_DATE),
1232                 'MMDDYYYY'),
1233         TO_CHAR(TO_DATE(RVHS_GB_POLY_EFF_DATE),
1234                 'YYYYMMDD'),
1235         ROUND(RVHS_WC_WEEKLY_WAGE,0),
1236         NVL(LPAD(DECODE(RVHS_POBI_CL_TYPE_CODE,'UK',
1237                         '0065',
1238                         RVHS_POBI_CL_TYPE_CODE),4,'0'),
1239              '0065'),
1240         NVL(LPAD(RVHS_NOSI_CL_TYPE_CODE,4,'0'),'0059'),
1241         RVHS_COLT_CL_TYPE_CODE,
1242         TO_CHAR(TO_DATE(RVHS_CL_REPORT_DATE),'MMDDYYYY'),
1243         RVHS_PLSM_GB_NO,
1244         RVHS_CLOC_CL_NO,
1245         RVHS_ULCT_ID_ACT ,
1246         RVHS_ULCT_ID_LOSS ,
1247         RVHS_ULCT_ID_RECV ,
1248         RVHS_ULCT_ID_COVG ,
1249         RVHS_ULCT_ID_SETT ,
1250         RVHS_ULCT_ID_FRAUD
1251 FROM   RVHS
1252 WHERE  ADD_MONTHS(LAST_DAY(RVHS_GB_POLY_EFF_DATE),127)
1253        > :WS-END-DATE
1254        AND RVHS_RSRV_CL_NO IS NOT NULL
1255        AND SUBSTR(RVHS_GB_POLY_EFF_DATE,4,3) = :WS-RPT-MONTH
1256        AND RVHS_GB_POL_TYPE           = 'W'
1257        AND RVHS_GB_POLY_EFF_DATE
1258        <= TO_DATE(:WS-END-DATE-18,'DD-MON-YYYY')
1259        AND (((NVL(RVHS_CL_CLOSE_DATE,:WS-END-DATE)
1260              >= NVL(RVHS_CL_UNIT_STATS_DATE,
1261                     TO_DATE(DECODE(MONTHS_BETWEEN(
1262                         TO_DATE(TO_CHAR(TO_DATE(:WS-END-DATE),
1263                         'DD-MON-YYYY'),'MON-YYYY'),'MON-YYYY'),
1264                         TO_DATE(TO_CHAR(RVHS_GB_POLY_EFF_DATE,
1265                         'MON-YYYY'),'MON-YYYY'))),

```

```

1266         18, :WS-BEG-DATE-18,
1267         30, :WS-BEG-DATE-12,
1268         42, :WS-BEG-DATE-12,
1269         54, :WS-BEG-DATE-12,
1270         66, :WS-BEG-DATE-12,
1271         78, :WS-BEG-DATE-12,
1272         90, :WS-BEG-DATE-12,
1273         102, :WS-BEG-DATE-12,
1274         114, :WS-BEG-DATE-12,
1275         126, :WS-BEG-DATE-12), 'DD-MON-YYYY'))))
1276 OR (RVHS_CL_REOPEN_DATE IS NOT NULL
1277 AND RVHS_CL_REOPEN_DATE
1278 >= NVL(RVHS_CL_UNIT_STATS_DATE,
1279 TO_DATE(DECODE(MONTHS_BETWEEN(TO_DATE(TO_CHAR(
1280 TO_DATE(:WS-END-DATE, 'DD-MON-YYYY'),
1281 'MON-YYYY'), 'MON-YYYY'),
1282 TO_DATE(TO_CHAR(RVHS_GB_POLY_EFF_DATE,
1283 'MON-YYYY'), 'MON-YYYY')),
1284 18, :WS-BEG-DATE-18,
1285 30, :WS-BEG-DATE-12,
1286 42, :WS-BEG-DATE-12,
1287 54, :WS-BEG-DATE-12,
1288 66, :WS-BEG-DATE-12,
1289 78, :WS-BEG-DATE-12,
1290 90, :WS-BEG-DATE-12,
1291 102, :WS-BEG-DATE-12,
1292 114, :WS-BEG-DATE-12,
1293 126, :WS-BEG-DATE-12), 'DD-MON-YYYY'))))
1294 OR (RVHS_CL_UNIT_STATS_DATE IS NOT NULL
1295 AND RVHS_CL_NO IN
1296 (SELECT LSRV_RVHS_CL_NO
1297 FROM LSRV
1298 WHERE LSRV_GB_CNTRL_DATE
1299 = :WS-END-DATE))
1300 OR (RVHS_CL_UNIT_STATS_DATE IS NOT NULL
1301 AND RVHS_CL_NO IN
1302 (SELECT LSPD_RVHS_CL_NO
1303 FROM LSPD
1304 WHERE LSPD_DRTP_GB_TYPE IN
1305 ('RSAL', 'RSUB', 'RCON',
1306 'CDPM', 'CDPP', 'CKCP',
1307 'CKPM'))
1308 AND LSPD_CL_PAID_DATE
1309 BETWEEN
1310 TO_DATE(:WS-BEG-DATE-11, 'DD-MON-YYYY')
1311 AND :WS-END-DATE)))
1312 AND RVHS_GB_POLY_EFF_DATE > '31-JAN-1991')
1313 *****--PLI32502.11594
1314 OR (ADD_MONTHS(LAST_DAY(RVHS_GB_POLY_EFF_DATE), 127)
1315 > :WS-END-DATE
1316 AND MONTHS_BETWEEN(:WS-END-DATE,
1317 LAST_DAY(RVHS_GB_POLY_EFF_DATE)) >= 18
1318 AND SUBSTR(RVHS_GB_POLY_EFF_DATE, 4, 3) = :WS-RPT-MONTH
1319 AND RVHS_LSPD_DATE_CHK = :WS-END-DATE)
1320 *****--PLI32502.11594
1321 ORDER BY RVHS_BSOR_GB_FM_ACCT,

```

```

1322         RVHS_GB_STATE,
1323         RVHS_GB_POL_NUMBER,
1324         RVHS_CL_CLAIMN_NO,
1325         RVHS_RSTP_CL_TYPE_CODE
1326     END-EXEC.
1327 *
1328 * THIS CURSOR IS OPENED IF THE RERUN PARM IS USED
1329 *
1330 EXEC SQL AT ORA1
1331     DECLARE CURS_WC_LOSS_RR CURSOR FOR
1332         SELECT RVHS_CL_NO,
1333         RVHS_RSRV_CL_NO,
1334         RVHS_BSOR_GB_ACCOUNT,
1335         RVHS_GB_POL_NUMBER,
1336         RVHS_GB_POL_PREFIX,
1337         LPAD(RVHS_GB_CLASS_CODE,4,'0'),
1338         RVHS_GB_STATE,
1339         RVHS_CL_OCCURRENCE_DATE,
1340         TO_CHAR(TO_DATE(RVHS_CL_OCCURRENCE_DATE),'YYYYMMDD'),
1341         TO_CHAR(TO_DATE(RVHS_CL_OCCURRENCE_DATE),'MMDDYYYY'),
1342         RVHS_CL_CLAIMN_NO,
1343         LTRIM(LTRIM(RVHS_CL_CLAIMN_NO,'W'),'0'),
1344         RVHS_RSTP_CL_TYPE_CODE,
1345         DECODE (TO_NUMBER (SUBSTR (RVHS_CLIC_CL_CODE, 1, 1)), 0,
1346             TO_NUMBER (SUBSTR (RVHS_CLIC_CL_CODE, 2, 1)),
1347             TO_NUMBER (SUBSTR (RVHS_CLIC_CL_CODE, 1, 1))),
1348         TO_CHAR(TO_DATE(RVHS_CL_CLOSE_DATE),'YYYYMMDD'),
1349         TO_CHAR(TO_DATE(RVHS_CL_REOPEN_DATE),'YYYYMMDD'),
1350         RVHS_WCCC_CL_TYPE_CODE,
1351         RVHS_WCCC_CL_TYPE_CODE,
1352         RVHS_GB_POLY_EFF_DATE,
1353         RVHS_GB_POLY_EXP_DATE,
1354         RVHS_BSOR_GB_FM_ACCT,
1355         SUBSTR(RVHS_GB_CLAIMANT,1,60),
1356         TO_CHAR(TO_DATE(RVHS_GB_POLY_EFF_DATE),'MMDDYYYY'),
1357         TO_CHAR(TO_DATE(RVHS_GB_POLY_EFF_DATE),'YYYYMMDD'),
1358         ROUND(RVHS_WC_WEEKLY_WAGE,0),
1359         NVL(LPAD(DECODE(RVHS_POBI_CL_TYPE_CODE,'UK','0065',
1360             RVHS_POBI_CL_TYPE_CODE),4,'0'),'0065'),
1361         NVL(LPAD(RVHS_NOSI_CL_TYPE_CODE,4,'0'),'0059'),
1362         RVHS_COLT_CL_TYPE_CODE,
1363         TO_CHAR(TO_DATE(RVHS_CL_REPORT_DATE),'MMDDYYYY'),
1364         RVHS_PLSM_GB_NO,
1365         RVHS_CLOC_CL_NO,
1366         RVHS_ULCT_ID_ACT ,
1367         RVHS_ULCT_ID_LOSS ,
1368         RVHS_ULCT_ID_RECV ,
1369         RVHS_ULCT_ID_COVG ,
1370         RVHS_ULCT_ID_SETT ,
1371         RVHS_ULCT_ID_FRAUD
1372     FROM RVHS
1373     WHERE   RVHS_CL_UNIT_STATS_DATE
1374           =   TO_DATE(:WS-RERUN-DATE,'DD-MON-YYYY')
1375     ORDER BY RVHS_BSOR_GB_FM_ACCT,
1376             RVHS_GB_STATE,
1377             RVHS_GB_POL_NUMBER,

```

```

1378         RVHS_CL_CLAIMN_NO,
1379         RVHS_RSTP_CL_TYPE_CODE
1380     END-EXEC.
1381 *
1382     EXEC SQL AT ora1
1383         DECLARE GET_RPT_NO CURSOR FOR
1384             SELECT MAX(UNLS_CL_REPORT_NO)
1385             FROM unls
1386             WHERE UNLS_CL_CLAIM_NUMBER = :WS-RVHS-CL-CLAIM-NO-TRIM
1387     END-EXEC.
1388 *
1389     DISPLAY 'WS-RPT-MONTH ' ;      WS-RPT-MONTH.
1390     DISPLAY 'WS-BEG-DATE-11 ' ;   WS-BEG-DATE-11.
1391     DISPLAY 'WS-BEG-DATE-12 ' ;   WS-BEG-DATE-12.
1392     DISPLAY 'WS-BEG-DATE-18 ' ;   WS-BEG-DATE-18.
1393     DISPLAY 'WS-END-DATE-18 ' ;   WS-END-DATE-18.
1394     DISPLAY 'WS-END-DATE ' ;      WS-END-DATE.
1395     DISPLAY 'WS-BEG-DATE ' ;      WS-BEG-DATE.
1396     DISPLAY 'WS-DTCL-GB-END-DATE ' ; WS-DTCL-GB-END-DATE.
1397 *
1398     EXEC SQL  AT ORA1
1399         DECLARE LOSS_PAID_CURS CURSOR FOR
1400             SELECT RVHS_GB_CLASS_CODE,
1401                 ROUND(LSPD_CL_PAID_AMOUNT),
1402                 LSPD_DRTP_GB_TYPE,
1403                 LSPD_DFCM_CL_TYPE_CODE
1404             FROM  RVHS, LSPD
1405             WHERE LSPD_RVHS_CL_NO = RVHS_CL_NO
1406             AND   RVHS_CL_NO      = :WS-RVHS-CL-NO
1407             AND   LSPD_CL_PAID_DATE <=
1408                 TO_DATE(:WS-RERUN-DATE, 'DD-MON-YYYY')
1409             AND   LSPD_DRTP_GB_TYPE IN
1410                 ('RSAL','RSUB','RCON',
1411                 'CDPM','CDPP','CKCP','CKPM')
1412     END-EXEC.
1413 *
1414 *****
1415 * THIS BELOW WAS COMMENTED OUT IN KEVIN'S VERSION
1416 *****
1417     IF WS-RERUN = 'Y'
1418         PERFORM 0111-OPEN-RERUN THRU 0111-EXIT
1419         PERFORM 0113-OPEN-RPTNUM THRU 0113-EXIT
1420     ELSE
1421         PERFORM 0112-OPEN-STD  THRU 0112-EXIT
1422         PERFORM 0113-OPEN-RPTNUM THRU 0113-EXIT.
1423 *****
1424 * THIS ABOVE WAS COMMENTED OUT IN KEVIN'S VERSION
1425 *****
1426 *
1427     MOVE 0 TO WS-PAGE.
1428     MOVE 0 TO WS-MED-COUNT.
1429     MOVE 0 TO WS-INDEM-COUNT.
1430 *
1431 *****
1432 * THIS BELOW WAS COMMENTED OUT IN KEVIN'S VERSION
1433 *****

```



```

1434     IF WS-RERUN = 'Y'
1435         PERFORM 3005-FETCH-WCLOSS-RR THRU 3005-EXIT
1436     ELSE
1437         PERFORM 3000-FETCH-WCLOSS   THRU 3000-EXIT.
1438 *****
1439 * THIS ABOVE WAS COMMENTED OUT IN KEVIN'S VERSION
1440 *****
1441 *
1442     MOVE  WS-RVHS-GB-STATE TO WS-HOLD-STATE.
1443     PERFORM 7400-CHECK-STATE THRU 7400-EXIT.
1444     MOVE 0                TO WS-HOLD-BSOR-GB-FM-ACCT.
1445     MOVE -1                TO WS-HOLD-RVHS-RSRV-CL-NO.
1446     PERFORM 7000-PRINT-HEADING THRU 7000-EXIT.
1447 *
1448     EXEC SQL  AT ORA1
1449         SELECT TO_CHAR(SYSDATE,'MMDDYY'),
1450             TO_CHAR(SYSDATE,'MMDDYYYY'),
1451             TO_CHAR(SYSDATE,'YYDDD')
1452         INTO :WS-SYSDATE-MMDDYY,
1453             :WS-SYSDATE-MMDDYYYY,
1454             :WS-SYSDATE-YYDDD
1455         FROM   DUAL
1456     END-EXEC.
1457 *
1458     IF SQLCODE NOT EQUAL 0 THEN
1459         MOVE "*** ERROR ** GETTING SYSDATE INFO 0100-SR ** "
1460         TO WS-ERR-MSG
1461         PERFORM 9900-ABORT THRU 9900-EXIT.
1462 *
1463     PERFORM 8075-EXCEPTION-HEADINGS THRU 8075-EXIT.
1464 *
1465     0100-EXIT.
1466     EXIT.
1467 *
1468     0111-OPEN-RERUN.
1469 *
1470     DISPLAY 'OPEN CURS_WC_LOSS_RR'.
1471     EXEC SQL  AT ORA1
1472         OPEN CURS_WC_LOSS_RR
1473     END-EXEC.
1474 *
1475     IF SQLCODE NOT = 0 AND SQLCODE NOT = 1403
1476         MOVE "*** ERROR ** ON OPEN 0111"
1477         TO WS-ERR-MSG
1478         PERFORM 9900-ABORT THRU 9900-EXIT.
1479 *
1480     0111-EXIT.
1481     EXIT.
1482 *
1483 *
1484 *
1485     0112-OPEN-STD.
1486 *
1487
1488     DISPLAY 'OPEN CURS_WC_LOSS'.
1489     EXEC SQL AT ORA1

```

```

1490         OPEN CURS_WC_LOSS
1491     END-EXEC.
1492
1493     IF SQLCODE NOT = 0 AND SQLCODE NOT = 1403
1494         MOVE "*** ERROR ** OPEN 0112"
1495         TO WS-ERR-MSG
1496         PERFORM 9900-ABORT THRU 9900-EXIT.
1497
1498     *****-- PLI32502.16006
1499     *****-- Here we call the procedure to update the RVHS rows prior
1500     *****-- to processing
1501         display 'CALLING ID_LATE_CLAIMS'.
1502
1503     EXEC SQL AT ORA1
1504         CALL P_ID_LATE_CLAIMS(WS-RERUN-DATE)
1505     END-EXEC.
1506
1507
1508     *
1509     0112-EXIT.
1510     EXIT.
1511     *
1512     0113-OPEN-RPTNUM.
1513     *
1514         DISPLAY 'OPEN GET_RPT_NO'.
1515         EXEC SQL AT ORA1
1516             OPEN GET_RPT_NO
1517         END-EXEC.
1518     *
1519         IF SQLCODE NOT = 0 AND SQLCODE NOT = 1403
1520             MOVE "*** ERROR ** ON OPEN 0113"
1521             TO WS-ERR-MSG
1522             PERFORM 9900-ABORT THRU 9900-EXIT.
1523     *
1524     0113-EXIT.
1525     EXIT.
1526     *
1527     0110-GET-DATES.
1528     *
1529         EXEC SQL AT ORA1
1530             SELECT TO_CHAR(TO_DATE(ADD_MONTHS(DTCL_GB_END_DATE,-13)),
1531                 'DD-MON-YYYY'),
1532                 TO_CHAR(TO_DATE(ADD_MONTHS(DTCL_GB_END_DATE,-13)),
1533                 'YYYYMMDD'),
1534                 DTCL_GB_END_DATE,
1535                 DTCL_GB_END_DATE,
1536                 TO_CHAR(DTCL_GB_END_DATE,'YYYYMMDD'),
1537                 TO_CHAR(ADD_MONTHS(DTCL_GB_END_DATE,-18),'DD-MON-YYYY'),
1538                 TO_CHAR(ADD_MONTHS(DTCL_GB_BEG_DATE,-18),'DD-MON-YYYY'),
1539                 TO_CHAR(ADD_MONTHS(DTCL_GB_BEG_DATE,-12),'DD-MON-YYYY'),
1540                 TO_CHAR(ADD_MONTHS(DTCL_GB_BEG_DATE,-11),'DD-MON-YYYY'),
1541                 SUBSTR(ADD_MONTHS(DTCL_GB_END_DATE,-6),4,3),
1542                 DTCL_GB_BEG_DATE,
1543                 DTCL_GB_END_DATE
1544             INTO :WS-BEG-DATE,
1545                 :WS-BEG-DATE-YYYY,

```

```

1546      :WS-END-DATE,
1547      :WS-END-DATE-YY,
1548      :WS-END-DATE-YYYY,
1549      :WS-END-DATE-18,
1550      :WS-BEG-DATE-18,
1551      :WS-BEG-DATE-12,
1552      :WS-BEG-DATE-11,
1553      :WS-RPT-MONTH,
1554      :WS-DTCL-GB-BEG-DATE,
1555      :WS-DTCL-GB-END-DATE
1556 FROM DATE_CONTROL
1557 WHERE DTCL_GB_SYSTEM = 'CL'
1558 END-EXEC.
1559 *
1560 IF SQLCODE NOT EQUAL 0
1561     MOVE "*** ERROR SELECT DATE_CONTROL IN 0110-"
1562     TO WS-ERR-MSG
1563     PERFORM 9900-ABORT THRU 9900-EXIT.
1564 0110-EXIT.
1565 EXIT.
1566 *
1567 0120-FORMAT-DATES.
1568 *
1569     STRING '01-',
1570           WS-END-DATE(4:8)
1571           DELIMITED BY SIZE
1572           INTO WS-BEG-DATE.
1573 *
1574     STRING WS-END-DATE(1:7), WS-END-DATE(8:4)
1575           DELIMITED BY SIZE
1576           INTO WS-END-DATE-YY
1577 *
1578 *
1579 *
1580 EXEC SQL  AT ORA1
1581 SELECT
1582 TO_CHAR(ADD_MONTHS(TO_DATE(:WS-END-DATE,'DD-MON-YYYY'),-13),
1583 'DD-MON-YYYY'),
1584 TO_CHAR(TO_DATE(:WS-END-DATE-YY,'DD-MON-YYYY'),'YYYYMMDD'),
1585 TO_CHAR(ADD_MONTHS(TO_DATE(:WS-END-DATE,
1586 'DD-MON-YYYY'),-18),'DD-MON-YYYY'),
1587 TO_CHAR(ADD_MONTHS(TO_DATE(:WS-BEG-DATE,
1588 'DD-MON-YYYY'),-18),'DD-MON-YYYY'),
1589 TO_CHAR(ADD_MONTHS(TO_DATE(:WS-BEG-DATE,
1590 'DD-MON-YYYY'),-12),'DD-MON-YYYY'),
1591 TO_CHAR(ADD_MONTHS(TO_DATE(:WS-BEG-DATE,
1592 'DD-MON-YYYY'),-11),'DD-MON-YYYY'),
1593 SUBSTR(ADD_MONTHS(TO_DATE(:WS-END-DATE,
1594 'DD-MON-YYYY'),-6),4,3),
1595 TO_CHAR(SYSDATE,'MM/DD/YYYYHH:MI:SS')
1596 INTO
1597     :WS-BEG-DATE,
1598     :WS-END-DATE-YYYY,
1599     :WS-END-DATE-18,
1600     :WS-BEG-DATE-18,
1601     :WS-BEG-DATE-12,

```

```

1602         :WS-BEG-DATE-11,
1603         :WS-RPT-MONTH,
1604         :WS-SYSDATE-TIME
1605     FROM DUAL
1606 END-EXEC.
1607 *
1608     IF SQLCODE NOT EQUAL 0
1609     MOVE "*** ERROR FORMATTING DATES IN 0120-"
1610     TO WS-ERR-MSG
1611     DISPLAY SQLCODE
1612     PERFORM 9900-ABORT THRU 9900-EXIT.
1613 *
1614 0120-EXIT.
1615     EXIT.
1616 *
1617 *-----|1000-PROCESS|-----*
1618 *-----*
1619 1000-PROCESS-REPORT.
1620
1621 *
1622     IF WS-HOLD-STATE NOT = WS-RVHS-GB-STATE THEN
1623     PERFORM 2500-STATE-BREAK THRU 2500-EXIT.
1624 *
1625
1626
1627
1628     PERFORM 1500-GET-RESERVES THRU 1500-EXIT.
1629     PERFORM 1100-GET-TID THRU 1100-EXIT.
1630     PERFORM 1110-FORMAT-TID THRU 1110-EXIT.
1631 *
1632     PERFORM 1800-GET-CA-FIELDS THRU 1800-EXIT.
1633 *
1634     PERFORM 2000-PRINT-DETAIL THRU 2000-EXIT.
1635     PERFORM 4000-OUTPUT-NCCI-DATA THRU 4000-EXIT.
1636 *
1637     IF WS-RERUN = 'Y'
1638     PERFORM 3005-FETCH-WCLOSS-RR THRU 3005-EXIT
1639     ELSE
1640     PERFORM 3000-FETCH-WCLOSS THRU 3000-EXIT.
1641 *
1642 1000-EXIT.
1643     EXIT.
1644 *
1645 1100-GET-TID.
1646 *
1647 *-----*
1648 *          1100-GET-TID          *
1649 *-----*
1650 *
1651     MOVE SPACES TO WS-CLPT-CL-TAX-ID-NO,
1652     WS-CLPT-CL-TAX-ID-TYPE,
1653     WS-TID,
1654     WS-CLPT-PTTP-CL-TYPE-CODE.
1655     MOVE ZEROES TO WS-UNLS-CL-AVG-WKLY-WAGE,
1656     WS-CLPT-CL-PERM-DISABL-RATING,
1657     WS-RSRV-CL-REHAB.

```

```

1658 *
1659
1660 *
1661 EXEC SQL AT ORA1
1662     SELECT NVL(CLPT_CL_TAX_ID_NO,'000000000'),
1663            CLPT_CL_TAX_ID_TYPE,
1664            CLPT_PTPP_CL_TYPE_CODE,
1665            SUBSTR(CLPR_CL_JOB_TITLE,1,18),
1666            ROUND(NVL(CLPR_CL_WC_WEEKLY_WAGES,0),0),
1667            NVL(CLPR_CL_PERM_DISABL_RATING,0),
1668            TO_CHAR(CLPT_CL_HIRE_DATE,'DDMONYYYY')
1669 INTO   :WS-CLPT-CL-TAX-ID-NO,
1670        :WS-CLPT-CL-TAX-ID-TYPE,
1671        :WS-CLPT-PTTP-CL-TYPE-CODE,
1672        :WS-UNLS-CLM-OCCUP-DESC,
1673        :WS-UNLS-CL-AVG-WKLY-WAGE,
1674        :WS-CLPT-CL-PERM-DISABL-RATING,
1675        :WS_CLPT_CL_HIRE_DATE
1676 FROM   CLPT,PTRL,
1677        CLPR,RSRV
1678 WHERE  CLPT_CL_NO           = PTRL_CLPT_CL_NO
1679        AND PTRL_CLPR_CL_NO   = CLPR_CL_NO
1680        AND CLPR_CL_NO        = RSRV_CLPR_CL_NO
1681        AND RSRV_CL_NO        IN
1682        (select MAX(RSRV_CL_NO)
1683         from   RSRV
1684         where  RSRV_CLOC_CL_CLAIMNO
1685               = :WS-RVHS-CL-CLAIM-NO
1686        AND    (RSRV_RSTP_CL_TYPE_CODE = 'WI' OR
1687               RSRV_RSTP_CL_TYPE_CODE = 'WM' ))
1688 *   need only most recent data = :WS-RVHS-RSRV-CL-NO
1689
1690 END-EXEC.
1691 *
1692 IF SQLCODE = 1403
1693 OR WS-CLPT-CL-TAX-ID-NO = '000000000'
1694     MOVE ' NONE' TO WS-TID
1695     MOVE 'N'     TO WS-CLPT-CL-TAX-ID-TYPE
1696 ELSE
1697     IF SQLCODE EQUAL 0
1698         NEXT SENTENCE
1699     ELSE
1700         MOVE   "*** ERROR ** ON SELECTING TID -1100"
1701             TO WS-ERR-MSG
1702         DISPLAY "*** ERROR ** ON SELECTING TID -1100"
1703             SQLCODE
1704         PERFORM 9900-ABORT THRU 9900-EXIT.
1705
1706 EXEC SQL AT ORA1
1707     SELECT
1708         NVL(RSRV_CL_REHAB,0)
1709 INTO
1710     :WS-RSRV-CL-REHAB
1711 FROM   CLPT,PTRL,
1712        CLPR,RSRV
1713 WHERE  CLPT_CL_NO           = PTRL_CLPT_CL_NO

```

```

1714         AND PTRL_CLPR_CL_NO      = CLPR_CL_NO
1715         AND CLPR_CL_NO            = RSRV_CLPR_CL_NO
1716         AND RSRV_CL_NO           IN
1717         (select MAX(RSRV_CL_NO)
1718          from   RSRV
1719          where  RSRV_CLOC_CL_CLAIMNO
1720                = :WS-RVHS-CL-CLAIM-NO
1721                AND  RSRV_RSTP_CL_TYPE_CODE
1722                = 'WI')
1723 *      need only most recent indemnity for rehab
1724
1725     END-EXEC.
1726 *
1727     IF SQLCODE = 1403
1728         MOVE 0 TO WS-RSRV-CL-REHAB
1729     ELSE
1730         IF SQLCODE EQUAL 0
1731             NEXT SENTENCE
1732         ELSE
1733             MOVE  '*** ERROR ** ON SELECTING REHAB -1100b'
1734                 TO WS-ERR-MSG
1735             DISPLAY '*** ERROR ** ON SELECTING REHAB -1100b'
1736                 SQLCODE
1737             PERFORM 9900-ABORT THRU 9900-EXIT.
1738 *
1739     1100-EXIT.
1740     EXIT.
1741 *
1742 * -----*
1743 *      1110-FORMAT-TID      *
1744 * -----*
1745     1110-FORMAT-TID.
1746 *
1747     IF WS-CLPT-CL-TAX-ID-TYPE = 'F'
1748         PERFORM 1115-FORMAT-F
1749     ELSE
1750         IF WS-CLPT-CL-TAX-ID-TYPE = 'S'
1751             PERFORM 1120-FORMAT-S
1752         ELSE
1753             NEXT SENTENCE.
1754 *
1755     1110-EXIT.
1756     EXIT.
1757 *
1758 * -----*
1759 *      1115-FORMAT-F      *
1760 * -----*
1761 *
1762     1115-FORMAT-F.
1763 *
1764     STRING WS-CLPT-CL-TAX-ID-NO(1:2)
1765           DELIMITED BY SIZE INTO WS-TID-F2.
1766     STRING WS-CLPT-CL-TAX-ID-NO(3:7)
1767           DELIMITED BY SIZE INTO WS-TID-F7.
1768     MOVE WS-TID-F TO WS-TID.
1769 *

```



```

1770 1115-EXIT.
1771 EXIT.
1772 *-----*
1773 *           1120-FORMAT-S           *
1774 *-----*
1775 *
1776 1120-FORMAT-S.
1777 *
1778     STRING WS-CLPT-CL-TAX-ID-NO(1:3)
1779           DELIMITED BY SIZE INTO WS-TID-S3.
1780     STRING WS-CLPT-CL-TAX-ID-NO(4:2)
1781           DELIMITED BY SIZE INTO WS-TID-S2.
1782     STRING WS-CLPT-CL-TAX-ID-NO(6:4)
1783           DELIMITED BY SIZE INTO WS-TID-S4.
1784     MOVE WS-TID-S TO WS-TID.
1785 *
1786 1120-EXIT.
1787 EXIT.
1788 *-----*
1789 *           1500-GET-RESERVES           *
1790 *-----*
1791 *
1792 1500-GET-RESERVES.
1793 *
1794     MOVE 0 TO WS-CURRENT-RESV.
1795 *
1796     EXEC SQL AT ORA1
1797     SELECT LSRV_CL_RESV_AMOUNT
1798     INTO :WS-CURRENT-RESV
1799     FROM LOSS_RESV
1800     WHERE LSRV_GB_CNTRL_DATE = :WS-END-DATE
1801     AND   LSRV_RVHS_CL_NO = :WS-RVHS-CL-NO
1802     END-EXEC.
1803 *
1804     IF SQLCODE NOT EQUAL 0 AND NOT EQUAL 1403
1805       MOVE "*** ERROR ** ON SELECTING LSRV -1500"
1806         TO WS-ERR-MSG
1807       PERFORM 9900-ABORT THRU 9900-EXIT.
1808 *
1809     MOVE 0 TO WS-PAID.
1810 *
1811     IF WS-RVHS-CL-OCCURRENCE-YYYY < '19880101'
1812       PERFORM 1600-SUM-DRAFT-DETAIL THRU 1600-EXIT
1813     ELSE
1814       PERFORM 1700-SUM-LOSS-PAID   THRU 1700-EXIT.
1815 *
1816 1500-EXIT.
1817 EXIT.
1818 *
1819 1600-SUM-DRAFT-DETAIL.
1820 *~~~~~
1821 *~~~~~
1822 * THE 1600 BLOCK IS CALLED ONLY WHEN THE OCCURRENCE DATE IS
1823 * EARLIER THAN 1988. POST-1988 DATES ARE HANDLED BY THE
1824 * 1700 BLOCK BELOW.
1825 *****

```

```

1826 *****
1827
1828 *
1829     EXEC SQL AT ORA1
1830         SELECT ROUND(SUM(DRFT_CL_AMOUNT))
1831         INTO :WS-PAID
1832         FROM DRAFT_DETAIL, DRFR
1833         WHERE DRFT_DRFR_CL_NO = DRFR_CL_NO
1834         AND DRFT_DRTP_CL_TYPE
1835             IN ('RSAL','RSUB','RCON', 'CDPM','CDPP','CKCP','CKPM')
1836         AND DRFR_DFCM_TYPE_CODE <> 'DED'
1837         AND DRFT_MCTB_CL_STAT_IND IN ('C','I','R')
1838         AND DRFT_RSRV_CL_NO = :WS-RVHS-RSRV-CL-NO
1839         AND DRFT_CL_CONTROL_DATE <= :WS-END-DATE
1840     END-EXEC.
1841
1842 *****
1843 * THIS IS THE CODE AS IT ORIGINALLY APPEARED IN THE PROGRAM.
1844 * IT IS THE BASIS FOR THE LOGIC ABOVE.
1845 *****
1846 *     EXEC SQL AT ORA1
1847 *         SELECT LTRIM(ROUND(SUM(DRFT_CL_AMOUNT)),0)
1848 *         INTO :WS-PAID
1849 *         FROM DRAFT_DETAIL
1850 *         WHERE DRFT_DRTP_CL_TYPE IN
1851 *             ('RSAL','RSUB','RCON',
1852 *              'CDPM','CDPP','CKCP','CKPM')
1853 *         AND DRFT_MCTB_CL_STAT_IND NOT IN('V','P','S')
1854 *         AND DRFT_RSRV_CL_NO = :WS-RVHS-RSRV-CL-NO
1855 *         AND DRFT_CL_CONTROL_DATE <= :WS-END-DATE
1856 *     END-EXEC.
1857 *****
1858
1859     IF SQLCODE NOT = 0 AND SQLCODE NOT = 1403
1860         MOVE "*** ERROR ** SUMMING DRAFT_DETAIL -1600"
1861         TO WS-ERR-MSG
1862         PERFORM 9900-ABORT THRU 9900-EXIT.
1863 *
1864
1865 1600-EXIT.
1866     EXIT.
1867 *
1868 * ~~~~~
1869 * ~~~~~
1870
1871 1700-SUM-LOSS-PAID.
1872 * ~~~~~
1873 * ~~~~~
1874
1875 * -----*
1876 * THIS SQL CALCULATES THE TOTAL AMOUNT OF THE LOSS BY EXCLUDING *
1877 * PAYMENTS (FROM THE CUSTOMER TO HORTICA) WHICH APPEAR AS *
1878 * CREDITS AND SKEW THE RESULT. WS-GROSS-PAID IS THE GROSS PAID *
1879 * AMOUNT WITHOUT ANY CREDITS APPLIED. *
1880 * -----*
1881

```

```

1882 EXEC SQL AT ORA1
1883 SELECT
1884 NVL(LTRIM(ROUND(SUM(LSPD_CL_PAID_AMOUNT)),0),0)
1885 INTO :WS-GROSS-PAID
1886 FROM LOSS_PAID
1887 WHERE LSPD_DRTP_GB_TYPE IN
1888 ('RSAL','RSUB','RCON','CDPM','CDPP','CKCP','CKPM')
1889 AND NVL(LSPD_DFCM_CL_TYPE_CODE,'SB3') <> 'DED'
1890 AND LSPD_CL_PAID_DATE
1891 <= TO_DATE(:WS-RERUN-DATE,'DD-MON-YYYY')
1892 AND LSPD_RVHS_CL_NO IN
1893 (SELECT RVHS_CL_NO
1894 FROM RVHS
1895 WHERE RVHS_CL_CLAIMN_NO = :WS-RVHS-CL-CLAIM-NO
1896 AND RVHS_RSTP_CL_TYPE_CODE = :WS-RVHS-RSTP-CL-TYPE-CODE)
1897 END-EXEC.
1898 *
1899 IF SQLCODE NOT = 0 AND SQLCODE NOT = 1403
1900 MOVE "*** ERROR ** SUMMING GROSS LOSS_PAID -1700"
1901 TO WS-ERR-MSG
1902 PERFORM 9900-ABORT THRU 9900-EXIT.
1903 *-----*
1904 * THIS SQL INCLUDES THE DEDUCTIBLES AND IS WHAT WAS ORIGINALLY *
1905 * USED FOR THE REPORT. WS-PAID IS THE NET AMOUNT PAID *
1906 * (IE: GROSS PAID AMOUNT MINUS ANY AMOUNTS CODED AS A DEDUCTIBLE *
1907 * ('DED')). *
1908 *-----*
1909
1910 EXEC SQL AT ORA1
1911 SELECT
1912 NVL(LTRIM(ROUND(SUM(LSPD_CL_PAID_AMOUNT)),0),0)
1913 INTO :WS-PAID
1914 FROM LOSS_PAID
1915 WHERE LSPD_DRTP_GB_TYPE IN
1916 ('RSAL','RSUB','RCON',
1917 'CDPM','CDPP','CKCP','CKPM')
1918 AND LSPD_CL_PAID_DATE
1919 <= TO_DATE(:WS-RERUN-DATE,'DD-MON-YYYY')
1920 AND LSPD_RVHS_CL_NO IN
1921 (SELECT RVHS_CL_NO
1922 FROM RVHS
1923 WHERE RVHS_CL_CLAIMN_NO = :WS-RVHS-CL-CLAIM-NO
1924 AND RVHS_RSTP_CL_TYPE_CODE = :WS-RVHS-RSTP-CL-TYPE-CODE)
1925 END-EXEC.
1926 *
1927 IF SQLCODE NOT = 0 AND SQLCODE NOT = 1403
1928 MOVE "*** ERROR ** SUMMING NET LOSS_PAID -1700"
1929 TO WS-ERR-MSG
1930 PERFORM 9900-ABORT THRU 9900-EXIT.
1931 *
1932 *
1933 PERFORM 1705-CALCULATE-DEDUCTIBLE THRU 1705-EXIT.
1934
1935 *****
1936
1937 EXEC SQL AT ORA1

```

```

1938         OPEN LOSS_PAID_CURS
1939     END-EXEC.
1940 *
1941     INITIALIZE WS-SUBRO-CHECK, WS-SIF-CHECK,
1942         WS-COV-B-CHECK, WS-COV-A-CHECK.
1943
1944     PERFORM 1750-FETCH-LOSS-PAID THRU 1750-EXIT UNTIL LOSS-EOF
1945 *
1946     IF WS-UNLS-LOSS-COVG = SPACES
1947     IF WS-RVHS-GB-STATE = '32'
1948         MOVE "00" TO WS-UNLS-LOSS-COVG
1949     ELSE
1950         MOVE "01" TO WS-UNLS-LOSS-COVG
1951     END-IF
1952 *
1953     IF WS-UNLS-LOSS-SETL = SPACES
1954     PERFORM 1770-GET-TYPE-OF-SETTLEMENT THRU 1770-EXIT
1955     END-IF
1956
1957 *****-- IF WS-UNLS-LOSS-RECOV = SPACES
1958     PERFORM 1780-MOVE-TYPE-OF-RECOV THRU 1780-EXIT.
1959 *****-- END-IF
1960
1961     PERFORM 1790-SUM-ALAE-AMOUNTS THRU 1790-EXIT
1962
1963
1964 *
1965     IF SQLCODE NOT EQUAL 0 THEN
1966         DISPLAY ">>>>>>>ERROR IN 1700-SUM-LOSS-PAID: " SQLCODE
1967     END-IF.
1968 *
1969     1700-EXIT.
1970     EXIT.
1971
1972 *BEGIN PLI31530.3248***
1973 *****
1974     1705-CALCULATE-DEDUCTIBLE.
1975
1976 *-----
1977 * THIS DETERMINES DEDUCTIBLE AVAILABILITY AND CALCULATES TOTAL
1978 * DEDUCTIBLE FOR THE POLICY. INDICATOR MARKERS ARE ALSO SET FOR
1979 * THE REPORT
1980 *-----
1981
1982 *-----
1983 * COMPUTE THE DEDUCTIBLE BASED ON VALUES FROM LOSS_PAID
1984 *-----
1985     COMPUTE WS-DEDUCT = (WS-GROSS-PAID - WS-PAID).
1986     MOVE WS-GROSS-PAID TO WS-PAID.
1987
1988 *-----
1989 * IDENTIFY POLICIES THAT HAVE DEDUCTIBLES BASED ON EXPOSURE
1990 * CLASS
1991 *-----
1992     MOVE -1 TO WS-WC-POL.
1993     MOVE SPACES TO WS-DEDUCT-IND.

```

```

1994 *
1995 EXEC SQL AT ORA1
1996 SELECT COUNT(POLY_GB_POL_NUMBER)
1997 INTO :WS-WC-POL
1998 FROM CLOC,
1999     POLY,
2000     INFC,
2001     COIT,
2002     EXIT,
2003     LOCT
2004 WHERE CLOC_POLY_GB_NO = POLY_GB_NO
2005 AND POLY_GB_NO = INFC_POLY_GB_NO_PART_OF
2006 AND INFC_GB_NO = COIT_INFC_GB_NO
2007 AND COIT_EXIT_GB_NO = EXIT_GB_NO
2008 AND COIT_LOCT_GB_NO = LOCT_GB_NO
2009 AND POLY_GB_POL_NUMBER = :WS-RVHS-POL-NO
2010 AND CLOC_CL_CLAIM_NO = :WS-RVHS-CL-CLAIM-NO
2011 AND LOCT_STAT_GB_ALPHA_STATE_IN LIKE :WS-RVHS-GB-STATE
2012 AND EXIT_WC_CLASS_CODE IN
2013 (SELECT BURT_WC_CLASS_CODE
2014  FROM BURT
2015   WHERE BURT_WC_STATE = :WS-RVHS-GB-STATE
2016   AND BURT_WC_DEDUCT_IND = 'Y')
2017 AND CLOC_CL_OCCURRENCE_DATE BETWEEN
2018     EXIT_GB_EFF_DATE AND EXIT_GB_EXP_DATE
2019 END-EXEC.
2020
2021
2022 *
2023 IF SQLCODE NOT EQUAL 0 THEN
2024 DISPLAY "ERROR DETERMINING CLASS CODE - 1705: " SQLCODE
2025 DISPLAY "RVHS POL: " WS-RVHS-POL-NO
2026 DISPLAY "RVHS CLAIM " WS-RVHS-CL-CLAIM-NO
2027 DISPLAY "RVHS STATE " WS-RVHS-GB-STATE
2028 END-IF.
2029
2030 *
2031 * -----
2032 * IF A POLICY IS SHOWN TO HAVE A DEDUCTIBLE, PRINT A 'Y' ON THE
2033 * REPORT. IF THE STATE IS FLORIDA OR IOWA, PRINT A '?' SINCE THERE
2034 * MAY OR MAY NOT BE A DEDUCTIBLE
2035 * -----
2036 IF WS-WC-POL > 0 THEN
2037     MOVE "Y" TO WS-DEDUCT-IND
2038     MOVE WS-DEDUCT-IND TO P-DED-IND
2039 ELSE
2040     MOVE 'N' TO WS-DEDUCT-IND
2041     MOVE WS-DEDUCT-IND TO P-DED-IND
2042 END-IF.
2043
2044 *
2045 IF WS-RVHS-GB-STATE = "FL" OR WS-RVHS-GB-STATE = "IA"
2046     THEN MOVE "?" TO WS-DEDUCT-IND
2047     MOVE WS-DEDUCT-IND TO P-DED-IND
2048 END-IF.
2049
2050 *
2051 * -----
2052 * CERTAIN STATES DO NOT OFFER A DEDUCTIBLE PROGRAM; IF THESE THERE

```

```

2050 * IS A DEDUCTIBLE AMOUNT SHOWN, THE POLICY SHOULD BE CHECKED FOR
2051 * POSSIBLE MISCODED TRANSACTIONS.
2052 * -----
2053 EXEC SQL AT ORA1
2054 SELECT 'N'
2055 INTO :WS-DEDUCT-IND
2056 FROM DUAL
2057 WHERE WS-RVHS-GB-STATE IN
2058 ('AK','AZ','AR','DC','ID',
2059 'LA','MS','OR','UT','VT',
2060 'WI','MI','NJ')
2061 END-EXEC.
2062 *
2063 IF WS-DEDUCT-IND = 'N' AND (WS-DEDUCT > 0
2064 OR WS-DEDUCT < 0) THEN
2065 MOVE '?' TO WS-DEDUCT-IND
2066 MOVE WS-DEDUCT-IND TO P-DED-IND
2067 ELSE
2068 MOVE WS-DEDUCT-IND TO P-DED-IND
2069 END-IF.
2070 * -----
2071 ** **
2072 * -----
2073 * DESIGNER CAPTURES WHAT THE DEDUCTIBLE "SHOULD BE." THE PROCEDURE
2074 * CALLED BELOW WILL RETURN THE DEDUCTIBLE AMOUNT AND LIMITATION.
2075 * IF THE DEDUCTIBLE AMOUNT IS GREATER THAN THE GROSS AMOUNT PAID,
2076 * THE GROSS AMOUNT WILL BE REPORTED AS THE DEDUCTIBLE AMOUNT
2077 ** **
2078 * -----
2079 IF WS-RVHS-POL-NO > 25000 OR WS-RVHS-POL-NO = 25000 THEN
2080 EXEC SQL AT ORA1 EXECUTE
2081 BEGIN
2082 P_GET_DEDUCTIBLE
2083 (WS-RVHS-POL-NO,
2084 'W',
2085 WS-RVHS-CL-OCC-DATE,
2086 WS-RVHS-GB-STATE,
2087 WS-DEDUCT,
2088 WS-MEANING,
2089 SQLCODE,
2090 WS-MESSAGE);
2091 END;
2092 END-EXEC
2093 IF WS-DEDUCT > WS-PAID THEN
2094 MOVE WS-PAID TO WS-DEDUCT
2095 MOVE WS-MEANING TO P-DESCRIPT
2096 END-IF
2097 END-IF.
2098 *
2099 MOVE WS-DEDUCT TO WS-UNLS-CL-DEDUCT-AMT.
2100 *
2101 1705-EXIT.
2102 EXIT.
2103 *****
2104 *END PLI31530.3248***
2105

```



```

2106 1750-FETCH-LOSS-PAID.
2107 * IF RECOVERIES EXIST FOR THE CLAIM, WE
2108 * WILL HAVE TO REPORT THAT THEY DO, NOT
2109 * HOW MUCH THOUGH - YET.
2110 *
2111 * RSUB HERE REPRESENTS WE HAVE RECIEVED
2112 * SOME MONIES FOR SUBROGATION FOR (THE ENTIRE CLAIM?)
2113 * THE CLAIM. BUT - HAVE TO SHOW IF WE GOT ANY
2114 * FROM THE SECOND INJURY FUND (DFCM_CL_TYPE_CODE = 'SIF')
2115 * AMONG OTHER THINGS...
2116 *****
2117 * DO NOT HAVE ANY WAY OF DETERMINING 5. OR 6. I.E. JOINT COVERAGE W/O
SUBRO
2118 *
2119 * AND JOINT COVERAGE W/ SUBRO
2120 * LOSS CONDITIONS - TYPE OF RECOVERY
2121 * -----
2122 * DESCRIPTION CODE
2123 *
2124 * 1. NO RECOVERY 01
2125 * 2. SECOND INJURY FUND ONLY 02
2126 * 3. SUBROGATION ONLY 03
2127 * 4. SUBROGATION AND SIF 04
2128 * 5. JOINT COVERAGE W/O SUBRO 05
2129 * 6. JOINT COVERAGE W/ SUBRO 06
2130 * THINK WILL HAVE TO MAKE CURSOR FOR CLASS CODES SINCE NEED TO KNOW IF
2131 * LOSSES WERE PAID OUT ON WHICH EXACT ONES, INSTEAD OF LUMPED ALL
TOGETHER
2132 *
2133 *
2134 *
2135 if WS-UNLS-LOSS-RECOV = spaces
2136 INITIALIZE WS-RVHS-GB-CLASS-CODE-ASWG,
2137 WS-LSPD-CL-PAID-AMOUNT-ASWG,
2138 WS-LSPD-DRTP-GB-TYPE-ASWG,
2139 WS-LSPD-DFCM-CL-TYPE-CODE-ASWG
2140 EXEC SQL AT ORA1
2141 FETCH LOSS_PAID_CURS
2142 INTO
2143 :WS-RVHS-GB-CLASS-CODE-ASWG,
2144 :WS-LSPD-CL-PAID-AMOUNT-ASWG,
2145 :WS-LSPD-DRTP-GB-TYPE-ASWG,
2146 :WS-LSPD-DFCM-CL-TYPE-CODE-ASWG
2147 END-EXEC
2148 *
2149 IF SQLCODE EQUAL 0
2150 PERFORM 1760-GET-TYPE-OF-RECOV THRU 1760-exit
2151 ELSE
2152 IF SQLCODE EQUAL 1403
2153 MOVE "Y" TO LOSS-EOF-SW
2154 ELSE
2155 MOVE "*** ERROR ** ON FETCH 1750" TO WS-ERR-MSG
2156 PERFORM 9900-ABORT THRU 9900-EXIT
2157 end-if
2158 end-if
2159 ELSE

```

```

2160         DISPLAY 'VALUE PRE-EXISTS'
2161         MOVE "Y" TO LOSS-EOF-SW
2162     END-IF.
2163 *****__*
2164     1750-EXIT.
2165     EXIT.
2166 *
2167     1760-GET-TYPE-OF-RECOV.
2168 *
2169         IF WS-LSPD-DRTP-GB-TYPE-ASWG = "RSUB"
2170             MOVE "Y" TO WS-SUBRO-CHECK
2171         ELSE
2172             IF WS-LSPD-DFCM-CL-TYPE-CODE-ASWG = "SIF"
2173                 MOVE "Y" TO WS-SIF-CHECK
2174             END-IF
2175         END-IF.
2176 *
2177 *
2178 *
2179 *
2180     1760-EXIT.
2181     EXIT.
2182 *
2183     1770-GET-TYPE-OF-SETTLEMENT.
2184 *
2185 * DON'T HAVE AWARD (03, 04) IN HERE YET. NOT CAPTURED IN
2186 * SYSTEM PER SAY, IT'S JUST MENTIONED IN THE CLAIM
2187 * COMMENTS...
2188 *
2189 * IF THERE IS NO DENIAL CODE AND PAYMENTS DON'T INVOLVE ANY TYPE
2190 * OF SETTLEMENT (DRAFT COMMENT DFCM = A1, ECT.)
2191 *
2192     INITIALIZE WS-RSRV-CLDC-CL-CODE-ASWG.
2193 *
2194     EXEC SQL AT ORA1
2195         SELECT RSRV_CLDC_CL_CODE
2196         INTO :WS-RSRV-CLDC-CL-CODE-ASWG
2197         FROM RVHS, RSRV
2198         WHERE RVHS_RSRV_CL_NO = RSRV_CL_NO
2199         AND RVHS_CL_NO = :WS-RVHS-CL-NO
2200     END-EXEC.
2201
2202     IF WS-RSRV-CLDC-CL-CODE-ASWG EQUAL SPACES
2203         AND ( WS-LSPD-DFCM-CL-TYPE-CODE-ASWG NOT EQUAL 'A1'
2204             OR WS-LSPD-DFCM-CL-TYPE-CODE-ASWG NOT EQUAL 'A2'
2205             OR WS-LSPD-DFCM-CL-TYPE-CODE-ASWG NOT EQUAL 'A3'
2206             OR WS-LSPD-DFCM-CL-TYPE-CODE-ASWG NOT EQUAL 'L7'
2207             OR WS-LSPD-DFCM-CL-TYPE-CODE-ASWG NOT EQUAL 'WC5')
2208             MOVE "00" TO WS-UNLS-LOSS-SETL
2209     ELSE
2210         IF (WS-RSRV-CLDC-CL-CODE-ASWG = "44"
2211             OR WS-RSRV-CLDC-CL-CODE-ASWG = "28")
2212             MOVE "05" TO WS-UNLS-LOSS-SETL
2213     ELSE
2214         IF WS-RSRV-CLDC-CL-CODE-ASWG = "24"
2215             IF WS-HOLD-STATE = 'NM'

```

```

2216         MOVE "07" TO WS-UNLS-LOSS-SETL
2217     ELSE
2218         IF WS-HOLD-STATE = 'NJ' THEN
2219             NEXT SENTENCE
2220         ELSE
2221             MOVE "09" TO WS-UNLS-LOSS-SETL
2222     ELSE
2223         IF WS-LSPD-DFCM-CL-TYPE-CODE-ASWG = "WC5"
2224             MOVE "06" TO WS-UNLS-LOSS-SETL
2225     ELSE
2226         IF WS-HOLD-STATE = 'CA'
2227             MOVE "01" TO WS-UNLS-LOSS-SETL
2228         else
2229             MOVE "00" TO WS-UNLS-LOSS-SETL.
2230 *
2231 1770-EXIT.
2232     EXIT.
2233 *
2234 1780-MOVE-TYPE-OF-RECOV.
2235 *
2236     IF WS-SUBRO-CHECK EQUAL SPACES AND WS-SIF-CHECK EQUAL SPACES
2237         MOVE "01" TO WS-UNLS-LOSS-RECOV
2238     END-IF.
2239
2240     IF WS-SUBRO-CHECK NOT EQUAL SPACES
2241         IF WS-SIF-CHECK NOT EQUAL SPACES
2242             MOVE "04" TO WS-UNLS-LOSS-RECOV
2243         ELSE
2244             MOVE "03" TO WS-UNLS-LOSS-RECOV
2245         END-IF
2246     ELSE
2247         IF WS-SIF-CHECK NOT EQUAL SPACES
2248             MOVE "02" TO WS-UNLS-LOSS-RECOV
2249         END-IF.
2250 *
2251 1780-EXIT.
2252     EXIT.
2253 *
2254 1790-SUM-ALAE-AMOUNTS.
2255 *
2256     INITIALIZE WS-UNLS-CLM-ALLOC-LAE-PD.
2257 *
2258 *BEGIN PLI31530.3248
2259 * Based on NCCI's 2003-2004 manual, deductibles are not
2260 * calculated into the ALAE amounts since we are to report the
2261 * total PAID ALAE
2262
2263     EXEC SQL AT ORA1
2264         SELECT ROUND(SUM(LSPD_CL_PAID_AMOUNT))
2265         INTO :WS-UNLS-CLM-ALLOC-LAE-PD
2266         FROM LSPD,RVHS
2267         WHERE LSPD_RVHS_CL_NO = RVHS_CL_NO
2268         AND LSPD_RVHS_CL_NO IN
2269         (SELECT RVHS_CL_NO
2270         FROM RVHS
2271         WHERE RVHS_CL_CLAIMN_NO = :WS-RVHS-CL-CLAIM-NO

```

```

2272         AND RVHS_RSTP_CL_TYPE_CODE = :WS-RVHS-RSTP-CL-TYPE-CODE)
2273         AND   LSPD_DRTP_GB_TYPE IN
2274             ('CDEP','CDPE','CKPE','CKEP')
2275         AND   LSPD_CL_PAID_DATE <=
2276             TO_DATE(:WS-RERUN-DATE, 'DD-MON-YYYY')
2277         AND   LSPD_DFCM_CL_TYPE_CODE <> 'DED'
2278     END-EXEC.
2279
2280     IF SQLCODE NOT EQUAL ZERO
2281         MOVE "*** ERROR ** ON FETCH 1790" TO WS-ERR-MSG
2282         PERFORM 9900-ABORT THRU 9900-EXIT
2283     END-IF.
2284 *END PLI31530.3248
2285
2286 *BEGIN PLI32502.11850
2287
2288     IF WS-UNLS-CLM-ALLOC-LAE-PD < 0 THEN
2289         MOVE 0 TO WS-UNLS-CLM-ALLOC-LAE-PD
2290     END-IF.
2291
2292 *END PLI32502.11850
2293 *begin PLI32502.15572
2294
2295
2296     IF WS-PAID < 0 THEN
2297         IF WS-RVHS-RSTP-CL-TYPE-CODE = 'WI'
2298             THEN
2299                 DISPLAY 'CLAIM:' WS-RVHS-CL-CLAIM-NO ' ' WS-RVHS-RSTP-CL-TYPE-CODE '
2300                     ' WS-UNLS-CLM-IDMNTY-PYMT-AMT ' ' WS-PAID
2301             END-IF
2302         IF WS-RVHS-RSTP-CL-TYPE-CODE = 'WM' THEN
2303             DISPLAY 'CLAIM:' WS-RVHS-CL-CLAIM-NO ' ' WS-RVHS-RSTP-CL-TYPE-CODE '
2304                 ' WS-UNLS-CLM-MED-PYMT-AMT ' ' WS-PAID
2305             END-IF
2306
2307         PERFORM 8050-PRINT-EXCEPTION-REPORT THRU 8050-EXIT
2308     END-IF.
2309
2310 *END PLI32502.15572
2311
2312 *
2313     1790-EXIT.
2314     EXIT.
2315 *****
2316     1800-GET-CA-FIELDS.
2317 *****
2318         MOVE ZEROES TO WS-UNLS-CL-PAID-VOC-REHAB,
2319             WS-SUBRO,
2320             WS-CLOC-CL-S-CLAIM,
2321             WS-CLOC-CL-JOINT-COV-PERCENT.
2322
2323     EXEC SQL AT ORA1
2324         SELECT ABS(ROUND(SUM(NVL(DRFT_CL_AMOUNT,0))))

```

```

2325         INTO :WS-UNLS-CL-PAID-VOC-REHAB
2326         FROM DRFT, DRFR, RSRV
2327         WHERE DRFT_MCTB_CL_STAT_IND NOT IN
2328             ('V','P','S')
2329         AND DRFR_DFCM_TYPE_CODE IN
2330             ('WVR', 'W1R', 'WIR', 'RSR', 'REF')
2331         AND RSRV_RSTP_CL_TYPE_CODE = 'WI'
2332         AND DRFT_DRTP_CL_TYPE IN
2333             ('RSAL','RSUB','RCON','CDPM','CDPP','CKCP','CKPM')
2334         AND DRFR_CL_NO = DRFT_DRFR_CL_NO
2335         AND DRFT_RSRV_CL_NO = RSRV_CL_NO
2336         AND DRFT_RSRV_CL_NO IN
2337             (select MAX(RSRV_CL_NO)
2338              from RSRV
2339              where RSRV_CLOC_CL_CLAIMNO = :WS-RVHS-CL-CLAIM-NO
2340              AND RSRV_RSTP_CL_TYPE_CODE = 'WI')
2341         END-EXEC.
2342 *
2343         IF SQLCODE NOT EQUAL 0 AND SQLCODE NOT EQUAL 1403
2344             MOVE "*** ERROR ** ON FETCH 1800" TO WS-ERR-MSG
2345             PERFORM 9900-ABORT THRU 9900-EXIT
2346         END-IF.
2347 *
2348         EXEC SQL AT ORA1
2349             SELECT ROUND(SUM(LSPD_CL_PAID_AMOUNT))
2350             INTO :WS-SUBRO
2351             FROM LOSS_PAID
2352             WHERE LSPD_DRTP_GB_TYPE = 'RSUB'
2353             AND LSPD_RVHS_CL_NO IN
2354                 (SELECT RVHS_CL_NO
2355                  FROM RVHS
2356                  WHERE RVHS_CL_CLAIMN_NO = :WS-RVHS-CL-CLAIM-NO)
2357             AND LSPD_CL_PAID_DATE
2358                 <= TO_DATE(:WS-RERUN-DATE,'DD-MON-YYYY')
2359         END-EXEC.
2360 *
2361         IF SQLCODE NOT = 0 AND SQLCODE NOT = 1403
2362             MOVE "*** ERROR ** SUMMING LOSS_PAID - RSUB -1800"
2363             TO WS-ERR-MSG
2364             PERFORM 9900-ABORT THRU 9900-EXIT.
2365 *
2366         EXEC SQL AT ORA1
2367             SELECT
2368                 NVL(CLOC_CL_S_CLAIM,0),
2369                 NVL(CLOC_CL_JOINT_COV_PERCENT,0)
2370         INTO
2371             :WS-CLOC-CL-S-CLAIM,
2372             :WS-CLOC-CL-JOINT-COV-PERCENT
2373         FROM CLOC
2374         WHERE CLOC_CL_CLAIM_NO = RTRIM(:WS-RVHS-CL-CLAIM-NO)
2375         END-EXEC.
2376 *
2377         IF SQLCODE = 1403
2378             DISPLAY "*** ERROR ** SR-4040 GETTING CLOC IN 1800-",
2379             DISPLAY 'CLAIMNO= ' WS-RVHS-CL-CLAIM-NO
2380         ELSE

```

```

2381         IF SQLCODE = 0
2382             NEXT SENTENCE
2383     ELSE
2384         DISPLAY "*** ERROR ** SR-4040 GETTING CLOC IN 1800-"
2385         PERFORM 9900-ABORT THRU 9900-EXIT.
2386
2387
2388     1800-EXIT.
2389     EXIT.
2390
2391 *-----*
2392 *                2000-PRINT-DETAIL                *
2393 *-----*
2394 *
2395     2000-PRINT-DETAIL.
2396 *
2397         ADD WS-CURRENT-RESV    TO WS-TOT-CUR-RESV.
2398 *BEGIN PLI31530.3248***
2399         ADD WS-GROSS-PAID      TO WS-TOT-PAID.
2400         ADD WS-DEDUCT          TO WS-TOT-DEDUCT.
2401 *END PLI31530.3248***
2402
2403 *
2404         IF WS-RVHS-RSTP-CL-TYPE-CODE = "WM" THEN
2405             ADD 1 TO WS-MED-COUNT
2406             ADD 1 TO WS-MED-TOTAL.
2407         IF WS-RVHS-RSTP-CL-TYPE-CODE = "WI" THEN
2408             ADD 1 TO WS-INDEM-COUNT
2409             ADD 1 TO WS-INDEM-TOTAL.
2410 *
2411         ADD 1 TO WS-TOTAL-CLAIMS.
2412         ADD 1 TO WS-ALL-CLAIMS.
2413 *
2414         IF WS-LINE-COUNT > 38 THEN
2415             PERFORM 7000-PRINT-HEADING THRU 7000-EXIT.
2416 *
2417         MOVE "O" TO P-STATUS.
2418 *
2419         IF WS-RVHS-CLOSE-DATE GREATER THAN WS-END-DATE-YYYY
2420         OR WS-RVHS-CLOSE-DATE = SPACES
2421             IF WS-RVHS-REOPEN-DATE = SPACES
2422                 MOVE 'O' TO P-STATUS
2423             ELSE
2424                 IF WS-RVHS-REOPEN-DATE NOT GREATER THAN
2425                     WS-END-DATE-YYYY
2426                     MOVE 'R' TO P-STATUS
2427                 ELSE
2428                     MOVE 'O' TO P-STATUS
2429             ELSE
2430                 MOVE 'C' TO P-STATUS.
2431
2432 *
2433         MOVE P-STATUS TO WS-UNLS-CLM-STATUS,
2434             CLAIM-STATUS.
2435
2436

```



```

2437         MOVE 'N' TO WS-STATUS-OVERRIDE.
2438
2439         EXEC SQL AT ORA1
2440         SELECT 'Y'
2441         INTO WS-STATUS-OVERRIDE
2442         FROM DUAL
2443         WHERE LAST_DAY (ADD_MONTHS (:WS-RVHS-GB-POLY-EFF-DATE, 18))
2444                = :WS-END-DATE
2445         END-EXEC.
2446
2447
2448
2449         IF WS-STATUS-OVERRIDE = 'Y'
2450             if WS-UNLS-CLM-STATUS = 'R'
2451                 MOVE 'O' TO WS-UNLS-CLM-STATUS
2452                 display 'claim status set to OPEN'
2453             END-IF
2454         END-IF.
2455
2456
2457
2458
2459
2460
2461 *****
2462 *BEGIN PLI32502.9994***
2463 *
2464 * THE LOGIC BELOW EXISTS BECAUSE THE CURRENT RESERVE ORIGINALLY
2465 * CALCULATED WAS: CR = ORIGINAL RESERVE - GROSS PAYMENTS + RECOVERY
2466 * (aka DEDUCTIBLE). THIS RESULTED IN A NET RESERVE FIGURE BEING
2467 * SHOWN ON THE REPORT AND CAPTURED FOR THE TABLE. THIS LOGIC
2468 * ONLY DEDUCTS THE RECOVERED AMOUNT AND RESULTS IN A GROSS RESERVE
2469
2470 *****--PLI32502.11531
2471 *****--
2472 *****-- IF P-STATUS = 'O'
2473 *****-- COMPUTE WS-CURRENT-RESV ROUNDED =
2474 *****-- (WS-CURRENT-RESV + WS-DEDUCT)
2475 *****-- END-IF.
2476 *****--
2477 *****--PLI32502.11531
2478
2479 *END PLI32502.9994
2480 *****
2481 *BEGIN PLI32502.7122
2482 ***** PLI32502.22151 -- IF WS-RVHS-GB-STATE = 'NJ'
2483 ***** PLI32502.22151 -- IF WS-RVHS-CLIC-CL-CODE = 3
2484 ***** PLI32502.22151 -- MOVE '9' TO WS-RVHS-CLIC-CL-CODE
2485 ***** PLI32502.22151 -- ELSE
2486 ***** PLI32502.22151 -- IF WS-RVHS-CLIC-CL-CODE = 4
2487 ***** PLI32502.22151 -- MOVE '9' TO WS-RVHS-CLIC-CL-CODE
2488 ***** PLI32502.22151 -- END-IF
2489 ***** PLI32502.22151 -- END-IF
2490 ***** PLI32502.22151 -- END-IF.
2491 *END PLI32502.7122
2492 *

```

```

2493      MOVE WS-RVHS-POL-NO          TO P-POL-NO.
2494      MOVE WS-RVHS-CLASS-CODE      TO P-CLASS-CODE.
2495      *      MOVE WS-RVHS-GB-STATE   TO P-STATE.
2496      MOVE WS-RVHS-CL-OCC-DATE      TO P-OCC-DATE.
2497      MOVE WS-RVHS-CL-CLAIM-NO      TO P-CLAIM-NO.
2498      MOVE WS-RVHS-RSTP-CL-TYPE-CODE TO P-RESV-TYPE.
2499      MOVE WS-RVHS-CLIC-CL-CODE     TO P-INJURY.
2500      MOVE WS-CURRENT-RESV          TO P-CURR-RESV
2501              WS-LSRV-CL-RESV-AMOUNT.
2502      MOVE WS-PAID                  TO P-PAID-AMT,
2503              WS-LSPD-CL-PAID-AMOUNT.
2504      MOVE WS-DEDUCT                TO P-DEDUCT-AMT.
2505      MOVE WS-RVHS-GB-POLY-EFF-DATE TO P-EFF-DATE.
2506      *
2507      IF SW-PROCESS-STATE-RPT2 = "Y" THEN
2508          WRITE PRINT-LINE2 FROM LINE-DETAIL
2509              AFTER ADVANCING 1 LINE
2510              MOVE SPACES TO PRINT-LINE2.
2511      *
2512      WRITE PRINT-LINE FROM LINE-DETAIL AFTER ADVANCING 1 LINE.
2513      MOVE SPACES TO PRINT-LINE.
2514      ADD +1 TO WS-LINE-COUNT.
2515      *
2516
2517      EXEC SQL AT ORA1
2518      SELECT DECODE(TO_CHAR(MONTHS_BETWEEN
2519          (:WS-END-DATE, LAST_DAY(:WS-RVHS-GB-POLY-EFF-DATE))),
2520          '18','1',
2521          '30','2',
2522          '42','3',
2523          '54','4',
2524          '66','5',
2525          '78','6',
2526          '90','7',
2527          '102','8',
2528          '114','9',
2529          '126','A',
2530          '139','B',
2531          'X') AS RPT_SEQ
2532      INTO WS-RPT-SEQ
2533      FROM DUAL
2534      END-EXEC.
2535
2536
2537      *****
2538      * THIS BELOW WILL NEED TO BE UNCOMMENTED IN KEVIN'S VERSION
2539      *****
2540      IF WS-RERUN = 'Y'
2541          NEXT SENTENCE
2542      ELSE
2543          PERFORM 7500-UPDATE-RESV-HIST THRU 7500-EXIT.
2544      *****
2545      * THIS ABOVE WILL NEED TO BE UNCOMMENTED IN KEVIN'S VERSION
2546      *****
2547      *
2548      2000-EXIT.

```

```

2549      EXIT.
2550      *
2551      *-----*
2552      *                2500-STATE-BREAK                *
2553      *-----*
2554      *
2555      2500-STATE-BREAK.
2556
2557      *
2558      MOVE WS-TOT-CUR-RESV  TO P-TOTAL-RESERVE.
2559      MOVE WS-TOT-PAID     TO P-TOTAL-PAID.
2560      MOVE WS-TOT-DEDUCT   TO P-TOTAL-DEDUCT.
2561      MOVE WS-MED-COUNT    TO P-TOTAL-MED.
2562      MOVE WS-INDEM-COUNT  TO P-TOTAL-INDEM.
2563      MOVE WS-TOTAL-CLAIMS TO P-TOTAL-CLAIMS.
2564      *
2565      PERFORM 6000-PRINT-TOTALS THRU 6000-EXIT.
2566      *
2567      ADD WS-TOT-CUR-RESV  TO WS-ALL-CUR-RESV.
2568      ADD WS-TOT-PAID     TO WS-ALL-PAID.
2569      ADD WS-TOT-DEDUCT   TO WS-ALL-DEDUCT.
2570      *
2571      MOVE WS-RVHS-GB-STATE TO WS-HOLD-STATE.
2572      *
2573      PERFORM 7400-CHECK-STATE THRU 7400-EXIT.
2574      *
2575      MOVE 0 TO WS-TOT-CUR-RESV.
2576      MOVE 0 TO WS-PAID.
2577      MOVE 0 TO WS-TOT-PAID.
2578      MOVE 0 TO WS-TOT-DEDUCT.
2579      MOVE 0 TO WS-MED-COUNT.
2580      MOVE 0 TO WS-INDEM-COUNT.
2581      MOVE 0 TO WS-TOTAL-CLAIMS.
2582      MOVE 46 TO WS-LINE-COUNT.
2583      *
2584      2500-EXIT.
2585      EXIT.
2586      *
2587      *-----*
2588      *                3000-FETCH-WCLOSS                *
2589      *-----*
2590      *
2591      3000-FETCH-WCLOSS.
2592
2593      *
2594      INITIALIZE
2595      WS-RVHS-CL-NO
2596      WS-RVHS-RSRV-CL-NO
2597      WS-RVHS-GB-ACCT
2598      WS-RVHS-POL-NO
2599      WS-RVHS-GB-POL-PREFIX
2600      WS-RVHS-CLASS-CODE
2601      WS-RVHS-GB-STATE
2602      WS-RVHS-CL-OCC-DATE
2603      WS-RVHS-CL-OCCURRENCE-YYYY
2604      WS-RVHS-CL-OCCURRENCE-MMDDYYYY

```

2605	WS-RVHS-CL-CLAIM-NO
2606	WS-RVHS-CL-CLAIM-NO-TRIM
2607	WS-RVHS-RSTP-CL-TYPE-CODE
2608	WS-RVHS-CLIC-CL-CODE
2609	WS-RVHS-CLOSE-DATE
2610	WS-RVHS-REOPEN-DATE
2611	WS-RVHS-WCCC-TYPE-CODE-C
2612	WS-RVHS-WCCC-TYPE-CODE-N
2613	WS-RVHS-GB-POLY-EFF-DATE
2614	WS-RVHS-GB-POLY-EXP-DATE
2615	WS-RVHS-BSOR-GB-FM-ACCT
2616	WS-RVHS-GB-CLAIMANT
2617	WS-RVHS-GB-POLY-EFF-MMDDYYYY
2618	WS-RVHS-GB-POLY-EFF-YYYYMMDD
2619	WS-RVHS-WC-WEEKLY-WAGE
2620	WS-RVHS-POBI-CL-TYPE-CODE
2621	WS-RVHS-NOSI-CL-TYPE-CODE
2622	WS-RVHS-COLT-CL-TYPE-CODE
2623	WS-RVHS-CL-REPORT-MMDDYYYY
2624	WS-RVHS-PLSM-GB-NO
2625	WS-RVHS-CLOC-CL-NO
2626	WS-RVHS-ACT-CODE
2627	WS-RVHS-LOSS-CODE
2628	WS-RVHS-RECV-CODE
2629	WS-RVHS-COVG-CODE
2630	WS-RVHS-SETT-CODE
2631	WS-RVHS-FRAUD-CODE.
2632	*
2633	EXEC SQL AT ORA1
2634	FETCH CURS_WC_LOSS
2635	INTO
2636	:WS-RVHS-CL-NO,
2637	:WS-RVHS-RSRV-CL-NO,
2638	:WS-RVHS-GB-ACCT,
2639	:WS-RVHS-POL-NO,
2640	:WS-RVHS-GB-POL-PREFIX,
2641	:WS-RVHS-CLASS-CODE,
2642	:WS-RVHS-GB-STATE,
2643	:WS-RVHS-CL-OCC-DATE,
2644	:WS-RVHS-CL-OCCURRENCE-YYYY,
2645	:WS-RVHS-CL-OCCURRENCE-MMDDYYYY,
2646	:WS-RVHS-CL-CLAIM-NO,
2647	:WS-RVHS-CL-CLAIM-NO-TRIM,
2648	:WS-RVHS-RSTP-CL-TYPE-CODE,
2649	:WS-RVHS-CLIC-CL-CODE,
2650	:WS-RVHS-CLOSE-DATE,
2651	:WS-RVHS-REOPEN-DATE,
2652	:WS-RVHS-WCCC-TYPE-CODE-C,
2653	:WS-RVHS-WCCC-TYPE-CODE-N,
2654	:WS-RVHS-GB-POLY-EFF-DATE,
2655	:WS-RVHS-GB-POLY-EXP-DATE,
2656	:WS-RVHS-BSOR-GB-FM-ACCT,
2657	:WS-RVHS-GB-CLAIMANT,
2658	:WS-RVHS-GB-POLY-EFF-MMDDYYYY,
2659	:WS-RVHS-GB-POLY-EFF-YYYYMMDD,
2660	:WS-RVHS-WC-WEEKLY-WAGE,

```

2661         :WS-RVHS-POBI-CL-TYPE-CODE,
2662         :WS-RVHS-NOSI-CL-TYPE-CODE,
2663         :WS-RVHS-COLT-CL-TYPE-CODE,
2664         :WS-RVHS-CL-REPORT-MMDDYYYY,
2665         :WS-RVHS-PLSM-GB-NO,
2666         :WS-RVHS-CLOC-CL-NO,
2667         :WS-RVHS-ACT-CODE,
2668         :WS-RVHS-LOSS-CODE,
2669         :WS-RVHS-RECV-CODE,
2670         :WS-RVHS-COVG-CODE,
2671         :WS-RVHS-SETT-CODE,
2672         :WS-RVHS-FRAUD-CODE
2673     END-EXEC.
2674
2675     *
2676     IF SQLCODE = 1403
2677         MOVE "Y" TO SW-EOF-CURS
2678     ELSE
2679         IF SQLCODE NOT = 0
2680             MOVE "*** ERROR ** ON FETCH 3000" TO WS-ERR-MSG
2681             display 'claim ' WS-RVHS-CL-CLAIM-NO
2682             PERFORM 9900-ABORT THRU 9900-EXIT.
2683     *     DISPLAY SQLCODE.
2684
2685
2686
2687         PERFORM 3010-ULCT-DECODE THRU 3010-EXIT.
2688
2689     3000-EXIT.
2690     EXIT.
2691
2692     *
2693     * -----*
2694     *           3005-FETCH-WCLOSS-RR           *
2695     * -----*
2696
2697     *
2698     3005-FETCH-WCLOSS-RR.
2699
2700     *
2701     INITIALIZE
2702     WS-RVHS-CL-NO
2703     WS-RVHS-RSRV-CL-NO
2704     WS-RVHS-GB-ACCT
2705     WS-RVHS-POL-NO
2706     WS-RVHS-GB-POL-PREFIX
2707     WS-RVHS-CLASS-CODE
2708     WS-RVHS-GB-STATE
2709     WS-RVHS-CL-OCC-DATE
2710     WS-RVHS-CL-OCCURRENCE-YYYY
2711     WS-RVHS-CL-OCCURRENCE-MMDDYYYY
2712     WS-RVHS-CL-CLAIM-NO
2713     WS-RVHS-CL-CLAIM-NO-TRIM
2714     WS-RVHS-RSTP-CL-TYPE-CODE
2715     WS-RVHS-CLIC-CL-CODE
2716     WS-RVHS-CLOSE-DATE
2717     WS-RVHS-REOPEN-DATE

```

2717	WS-RVHS-WCCC-TYPE-CODE-C
2718	WS-RVHS-WCCC-TYPE-CODE-N
2719	WS-RVHS-GB-POLY-EFF-DATE
2720	WS-RVHS-GB-POLY-EXP-DATE
2721	WS-RVHS-BSOR-GB-FM-ACCT
2722	WS-RVHS-GB-CLAIMANT
2723	WS-RVHS-GB-POLY-EFF-MMDDYYYY
2724	WS-RVHS-GB-POLY-EFF-YYYYMMDD
2725	WS-RVHS-WC-WEEKLY-WAGE
2726	WS-RVHS-POBI-CL-TYPE-CODE
2727	WS-RVHS-NOSI-CL-TYPE-CODE
2728	WS-RVHS-COLT-CL-TYPE-CODE
2729	WS-RVHS-CL-REPORT-MMDDYYYY
2730	WS-RVHS-PLSM-GB-NO
2731	WS-RVHS-CLOC-CL-NO
2732	WS-RVHS-ACT-CODE
2733	WS-RVHS-LOSS-CODE
2734	WS-RVHS-RECV-CODE
2735	WS-RVHS-COVG-CODE
2736	WS-RVHS-SETT-CODE
2737	WS-RVHS-FRAUD-CODE.
2738	*
2739	EXEC SQL AT ORA1
2740	FETCH CURS_WC_LOSS_RR
2741	INTO
2742	:WS-RVHS-CL-NO,
2743	:WS-RVHS-RSRV-CL-NO,
2744	:WS-RVHS-GB-ACCT,
2745	:WS-RVHS-POL-NO,
2746	:WS-RVHS-GB-POL-PREFIX,
2747	:WS-RVHS-CLASS-CODE,
2748	:WS-RVHS-GB-STATE,
2749	:WS-RVHS-CL-OCC-DATE,
2750	:WS-RVHS-CL-OCCURRENCE-YYYY,
2751	:WS-RVHS-CL-OCCURRENCE-MMDDYYYY,
2752	:WS-RVHS-CL-CLAIM-NO,
2753	:WS-RVHS-CL-CLAIM-NO-TRIM,
2754	:WS-RVHS-RSTP-CL-TYPE-CODE,
2755	:WS-RVHS-CLIC-CL-CODE,
2756	:WS-RVHS-CLOSE-DATE,
2757	:WS-RVHS-REOPEN-DATE,
2758	:WS-RVHS-WCCC-TYPE-CODE-C,
2759	:WS-RVHS-WCCC-TYPE-CODE-N,
2760	:WS-RVHS-GB-POLY-EFF-DATE,
2761	:WS-RVHS-GB-POLY-EXP-DATE,
2762	:WS-RVHS-BSOR-GB-FM-ACCT,
2763	:WS-RVHS-GB-CLAIMANT,
2764	:WS-RVHS-GB-POLY-EFF-MMDDYYYY,
2765	:WS-RVHS-GB-POLY-EFF-YYYYMMDD,
2766	:WS-RVHS-WC-WEEKLY-WAGE,
2767	:WS-RVHS-POBI-CL-TYPE-CODE,
2768	:WS-RVHS-NOSI-CL-TYPE-CODE,
2769	:WS-RVHS-COLT-CL-TYPE-CODE,
2770	:WS-RVHS-CL-REPORT-MMDDYYYY,
2771	:WS-RVHS-PLSM-GB-NO,
2772	:WS-RVHS-CLOC-CL-NO,


```

2773         :WS-RVHS-ACT-CODE,
2774         :WS-RVHS-LOSS-CODE,
2775         :WS-RVHS-RECV-CODE,
2776         :WS-RVHS-COVG-CODE,
2777         :WS-RVHS-SETT-CODE,
2778         :WS-RVHS-FRAUD-CODE
2779     END-EXEC.
2780 *
2781     IF SQLCODE = 0
2782         NEXT SENTENCE
2783     ELSE
2784         IF SQLCODE = 1403
2785             MOVE "Y" TO SW-EOF-CURS
2786         ELSE
2787             MOVE "*** ERROR ** ON FETCH 3005" TO WS-ERR-MSG
2788             PERFORM 9900-ABORT THRU 9900-EXIT.
2789
2790
2791         PERFORM 3010-ULCT-DECODE THRU 3010-EXIT.
2792 *
2793     3005-EXIT.
2794     EXIT.
2795
2796 *
2797 * -----*
2798 *           3010-ULCT-DECODE           *
2799 * -----*
2800 *
2801     3010-ULCT-DECODE.
2802
2803
2804         move spaces to WS-UNLS-LOSS-ACT.
2805         EXEC SQL AT ORA1
2806         SELECT ULCT_DATA_CODE ACT_CODE
2807         INTO :WS-UNLS-LOSS-ACT
2808         FROM   ULCT
2809         WHERE  ULCT_ID      = :WS-RVHS-ACT-CODE
2810         AND    ULCT_CODE_CATEGORY = 'ACT'
2811         END-EXEC.
2812
2813         move spaces to WS-UNLS-LOSS-TYPE.
2814         EXEC SQL AT ORA1
2815         SELECT ULCT_DATA_CODE LOSS_CODE
2816         INTO :WS-UNLS-LOSS-TYPE
2817         FROM   ULCT
2818         WHERE  ULCT_ID      = :WS-RVHS-LOSS-CODE
2819         AND    ULCT_CODE_CATEGORY = 'LOSS'
2820         END-EXEC.
2821
2822
2823         move spaces to WS-UNLS-LOSS-RECOV.
2824         EXEC SQL AT ORA1
2825         SELECT ULCT_DATA_CODE RECOV_CODE
2826         INTO :WS-UNLS-LOSS-RECOV
2827         FROM   ULCT
2828         WHERE  ULCT_ID      = :WS-RVHS-RECV-CODE

```

```

2829         AND  ULCT_CODE_CATEGORY = 'RECOVERY'
2830     END-EXEC.

2831
2832     move spaces to WS-UNLS-LOSS-COVG.
2833     EXEC SQL AT ORA1
2834     SELECT ULCT_DATA_CODE COVER_CODE
2835     INTO  :WS-UNLS-LOSS-COVG
2836     FROM  ULCT
2837     WHERE ULCT_ID      = :WS-RVHS-COVG-CODE
2838     AND   ULCT_CODE_CATEGORY = 'COVERAGE'
2839     END-EXEC.

2840
2841     move spaces to WS-UNLS-LOSS-SETL.
2842     EXEC SQL AT ORA1
2843     SELECT ULCT_DATA_CODE SETTLE_CODE
2844     INTO  :WS-UNLS-LOSS-SETL
2845     FROM  ULCT
2846     WHERE ULCT_ID      = :WS-RVHS-SETT-CODE
2847     AND   ULCT_CODE_CATEGORY = 'SETTLEMENT'
2848     END-EXEC.

2849
2850     move spaces to WS-UNLS-CLM-FRAUD-IND.
2851     EXEC SQL AT ORA1
2852     SELECT ULCT_DATA_CODE FRAUD_CODE
2853     INTO  :WS-UNLS-CLM-FRAUD-IND
2854     FROM  ULCT
2855     WHERE ULCT_ID      = :WS-RVHS-FRAUD-CODE
2856     AND   ULCT_CODE_CATEGORY = 'FRAUD'
2857     END-EXEC.

2858
2859 3010-EXIT.
2860     EXIT.

2861
2862 *
2863 *-----*
2864 *              4000-OUTPUT-NCCI-DATA              *
2865 *-----*
2866 *
2867 4000-OUTPUT-NCCI-DATA.
2868 *
2869     INITIALIZE CLAIM-FILLER.
2870     MOVE ZEROES      TO  CLAIM-LOCATION-NUMBER,
2871                      WS-UNLS-CLM-CAT-NO,
2872                      CLAIM-CATASTROPHE-NUMBER,
2873                      CLAIM-REHAB-RESERVE-AMT,
2874                      CLAIM-REHAB-PAYMENT-AMT,
2875                      WS-UNLS-CLM-CLAIMAINT-ATT-FEE,
2876                      CLAIM-CLAIMAINT-ATT-FEE,
2877                      WS-UNLS-CLM-EMPLOYER-ATT-FEE,
2878                      CLAIM-EMPLOYER-ATT-FEE,
2879 *                      WS-UNLS-CLM-ALLOC-LAE-PD,
2880                      CLAIM-ALLOC-LAE-PD,
2881                      WS-UNLS-CLM-ALLOC-LAE-IN,
2882                      CLAIM-ALLOC-LAE-IN,
2883                      WS-UNLS-CLM-DEDUCT-CODE,
2884                      CLAIM-DEDUCT-CODE.

```

```

2885 *
2886     MOVE SPACES      TO CLAIM-AGENT-CODE,
2887                     WS-UNLS-CLM-HMO-TYPE,
2888                     CLAIM-HMO-TYPE,
2889 *                     WS-UNLS-CLM-OCCUP-DESC,
2890                     CLAIM-OCCUP-DESC,
2891 *                     WS-UNLS-CLM-VOC-REHAB-IND,
2892                     CLAIM-VOC-REHAB-IND,
2893                     CLAIM-LUMP-SUM-IND,
2894                     CLAIM-FRAUD-IND.
2895     MOVE "N" TO      WS-UNLS-CLM-LUMP-SUM-IND.
2896
2897     IF WS-UNLS-CLM-FRAUD-IND = SPACES
2898         MOVE "00" TO      WS-UNLS-CLM-FRAUD-IND
2899     END-IF.
2900 *
2901     PERFORM 4010-STRING-POLICY-NUMBER THRU 4010-EXIT.
2902     PERFORM 4020-GET-EXIT THRU 4020-EXIT.
2903
2904 * GETTING THE EXPENSE AMOUNT HERE FOR FORMER NCCI DATA (AGO STUFF)
2905 * BERNIE SAYS THE PERCENT IS WRONG. CAG
2906
2907     IF WS-RVHS-RSTP-CL-TYPE-CODE EQUAL 'WI' OR 'WM'
2908         PERFORM 4090-GET-LSPD-STUFF THRU 4090-EXIT
2909         COMPUTE WS-EXPENSE-RESERVE ROUNDED =
2910             (WS-LSRV-CL-RESV-AMOUNT * .25).
2911     MOVE WS-EXPENSE-RESERVE  TO CLAIM-EXPENSE-RESERVE-AMT.
2912 *
2913 * LOAD THE CLAIM DATA
2914 *
2915     MOVE " "  TO CLAIM-EXPENSE-CODE.
2916     MOVE WS-NCCI-POLICY-NUMBER TO CLAIM-POLICY-NUMBER,
2917         CLAIM-POLNUM-POLNO.
2918 *
2919 *
2920 * AT SOME POINT WE WILL NEED TO DEVELOP A METHOD TO IDENTIFY OTHER FM
2921 * COMPANIES THAN 1. THE FIRST 4 POSITIONS OF THE CUSTOMER NUMBER
2922 * WILL BE 0001, AND THE LAST POSITION WILL BE A 1 DIGIT IDENTIFIER
2923 * NUMERIC, THAT WILL BE UNIQUE FOR THAT COMPANY.
2924 * *****
2925 * ***** NOTE!!!! NOTE!!!!
2926 * *****
2927 * WE ALSO MUST ADD THE COMPANY NCCI CARRIER CODE TO THE
2928 * AGO PC SYSTEM BY USING THE "MAINTAIN ASSOCIATION LIST"
2929 * SCREEN.
2930 *
2931     MOVE 0001 TO CLAIM-CUSTOMER-NUMBER-F4.
2932     MOVE 0  TO CLAIM-CUSTOMER-NUMBER-L1.
2933 * IF WS-RVHS-BSOR-GB-FM-ACCT < 10000
2934 *     MOVE WS-RVHS-BSOR-GB-FM-ACCT TO CLAIM-CUSTOMER-NUMBER-F4
2935 * ELSE
2936 *     MOVE WS-RVHS-BSOR-GB-FM-ACCT TO CLAIM-CUSTOMER-NUMBER.
2937 *
2938     MOVE WS-RVHS-GB-POLY-EFF-MMDDYYYY TO CLAIM-INCEPTION-DATE.
2939     MOVE WS-RVHS-CL-CLAIM-NO-TRIM  TO WS-UNLS-CLM-NUMBER,
2940         CLAIM-NUMBER.

```

```

2941 IF WS-RVHS-WC-WEEKLY-WAGE EQUAL ZEROES
2942     MOVE WS-UNLS-CL-AVG-WKLY-WAGE TO CLAIM-AVERAGE-WEEKLY-WAGE
2943 ELSE
2944     MOVE WS-RVHS-WC-WEEKLY-WAGE TO CLAIM-AVERAGE-WEEKLY-WAGE
2945         WS-UNLS-CL-AVG-WKLY-WAGE.
2946
2947 MOVE WS-RVHS-POBI-CL-TYPE-CODE TO WS-UNLS-CLM-BODY-PART-CDE
2948     CLAIM-BODY-PART-CODE.
2949
2950 INSPECT CLAIM-BODY-PART-CODE REPLACING LEADING ' ' BY '0'.
2951 MOVE WS-RVHS-NOSI-CL-TYPE-CODE TO
2952     WS-UNLS-CLM-NAT-OF-INJ-CDE,
2953     CLAIM-NATURE-OF-INJURY-CODE.
2954 INSPECT CLAIM-NATURE-OF-INJURY-CODE
2955     REPLACING LEADING ' ' BY '0'.
2956 MOVE WS-RVHS-COLT-CL-TYPE-CODE TO
2957     WS-UNLS-CLM-CSE-OF-INJ-CDE,
2958     CLAIM-CAUSE-OF-INJURY-CODE.
2959 MOVE SPACES TO CLAIM-LOSS-CODE-TYPE.
2960
2961
2962 MOVE WS-RVHS-CLIC-CL-CODE TO WS-UNLS-CLM-INJURY-CODE
2963     CLAIM-INJURY-CODE.
2964 MOVE WS-RVHS-COLT-CL-TYPE-CODE TO CLAIM-SOURCE-OF-INJURY.
2965 MOVE WS-RVHS-CL-OCC-DATE TO WS-UNLS-CLM-ACDNT-DTE.
2966 MOVE WS-RVHS-CL-OCCURRENCE-MMDDYYYY TO CLAIM-ACCIDENT-DATE.
2967 MOVE WS-RVHS-CL-REPORT-MMDDYYYY TO
2968     CLAIM-ORIGINAL-CLAIM-DATE.
2969 *
2970 MOVE WS-SYSDATE-MMDDYYYY TO CLAIM-TRANSACTION-DATE.
2971 MOVE P-STATUS TO WS-UNLS-CLM-STATUS,
2972     CLAIM-STATUS.
2973 MOVE WS-RVHS-CLASS-CODE TO CLAIM-CLASS-CODE.
2974
2975 IF WS-RVHS-RSTP-CL-TYPE-CODE EQUAL 'WI'
2976 THEN
2977     MOVE WS-LSRV-CL-RESV-AMOUNT TO CLAIM-INDEMNITY-RESERVE-AMT
2978     MOVE WS-LSRV-CL-RESV-AMOUNT TO CLAIM-INDEMNITY-RESERVE-AMT
2979     MOVE WS-LSPD-CL-PAID-AMOUNT TO WS-UNLS-CLM-IDMNTY-PYMT-AMT,
2980     CLAIM-INDEMNITY-PAYMENT-AMT
2981 MOVE ZEROES TO CLAIM-MEDICAL-RESERVE-AMT,
2982     WS-UNLS-CLM-MED-PYMT-AMT,
2983     CLAIM-MEDICAL-PAYMENT-AMT
2984 ELSE IF WS-RVHS-RSTP-CL-TYPE-CODE EQUAL 'WM'
2985 THEN
2986     MOVE WS-LSRV-CL-RESV-AMOUNT TO CLAIM-MEDICAL-RESERVE-AMT
2987     MOVE WS-LSPD-CL-PAID-AMOUNT TO WS-UNLS-CLM-MED-PYMT-AMT,
2988     CLAIM-MEDICAL-PAYMENT-AMT
2989 MOVE ZEROES TO CLAIM-INDEMNITY-RESERVE-AMT,
2990     WS-UNLS-CLM-IDMNTY-PYMT-AMT,
2991     CLAIM-INDEMNITY-PAYMENT-AMT.
2992 *
2993 MOVE WS-LSPD-EXP-PAYMENT-AMT TO CLAIM-EXPENSE-PAYMENT-AMT.
2994 MOVE WS-RVHS-WCCC-TYPE-CODE-C TO CLAIM-LOSS-COVERAGE-CODE.
2995 MOVE WS-RVHS-WCCC-TYPE-CODE-N TO WS-UNLS-COVG-CODE.
2996 MOVE WS-SYSDATE-TIME TO CLAIM-POLNUM-DT-TIME.

```

```

2997 *
2998     MOVE WS-RVHS-GB-STATE  TO CLAIM-STATE.
2999     MOVE WS-CLPT-CL-TAX-ID-NO  TO WS-UNLS-CLM-SSN
3000         CLAIM-SSN.
3001 ***** we no longer want to report SSN so move zeroes to all
3002 ***** 10/28/08: UNLESS IT'S CALIFORNIA
3003
3004     IF WS-RVHS-GB-STATE NOT = 'CA' THEN
3005     MOVE '000000000'          TO WS-UNLS-CLM-SSN
3006     END-IF.
3007
3008 *
3009 *****
3010 ** THE CODE BELOW EXISTS IN KEVIN'S VERSION.
3011 ** IT DOES NOT EXIST IN PRODUCTION VERSION.
3012 *****
3013 ** IF WE HAVE NO PAID OR RESERVE AMOUNT WE DON'T WANT TO WRITE THE
RECORD
3014 *
3015
3016     IF (WS-LSRV-CL-RESV-AMOUNT = 0
3017     AND WS-LSPD-CL-PAID-AMOUNT = 0)
3018
3019     NEXT SENTENCE
3020 ELSE
3021     IF WS-CREATE-DUMMY = 'Y'
3022     PERFORM 4005-WRITE-DUMMY THRU 4005-EXIT
3023     MOVE 'N' TO WS-CREATE-DUMMY
3024     MOVE WS-RVHS-CL-CLAIM-NO TO HOLD-CLAIM-NO.
3025 *     WRITE CLAIM-OUTPUT
3026 * ELSE
3027 *     WRITE CLAIM-OUTPUT.
3028 *
3029 *****
3030 ** THE CODE ABOVE WAS EXISTS IN KEVIN'S VERSION.
3031 ** IT DOES NOT EXIST IN PRODUCTION VERSION.
3032 *****
3033 *
3034 * WRITE CLAIM-OUTPUT.
3035 IF WS-UPDATE-SWITCH = 'Y'
3036     PERFORM 8000-GET-UNLS-DATA THRU 8000-EXIT
3037 ELSE
3038     NEXT SENTENCE.
3039 *
3040 4000-EXIT.
3041 EXIT.
3042
3043 *****
3044 ** THIS PROCEDURE CREATED BY KEVIN
3045 *****
3046 4005-WRITE-DUMMY.
3047 *
3048     MOVE CLAIM-OUTPUT TO HOLD-CLAIM-OUTPUT.
3049     MOVE 0 TO CLAIM-INDEMNITY-RESERVE-AMT
3050         WS-UNLS-CLM-IDMNTY-PYMT-AMT
3051         CLAIM-INDEMNITY-PAYMENT-AMT

```

```

3052 CLAIM-MEDICAL-RESERVE-AMT
3053 WS-UNLS-CLM-MED-PYMT-AMT
3054 CLAIM-MEDICAL-PAYMENT-AMT
3055 CLAIM-REHAB-RESERVE-AMT
3056 CLAIM-REHAB-PAYMENT-AMT
3057 CLAIM-EXPENSE-RESERVE-AMT
3058 CLAIM-EXPENSE-PAYMENT-AMT
3059 WS-UNLS-CLM-ALLOC-LAE-PD
3060 CLAIM-ALLOC-LAE-PD
3061 WS-UNLS-CLM-ALLOC-LAE-IN
3062 CLAIM-ALLOC-LAE-IN.
3063 *
3064 MOVE 'R' TO WS-UNLS-CLM-STATUS,
3065 CLAIM-STATUS.
3066 WRITE CLAIM-OUTPUT.
3067 MOVE HOLD-CLAIM-OUTPUT TO CLAIM-OUTPUT.
3068 *
3069 4005-EXIT.
3070 EXIT.
3071 *
3072 4010-STRING-POLICY-NUMBER.
3073 *
3074 *
3075 MOVE 0 TO WS-REWRITE-COUNT.
3076 MOVE SPACES TO WS-REWRITE,
3077 WS-NCCI-POLICY-NUMBER.
3078 *
3079 * IF THE POLICY WAS WRITTEN ON OR AFTER 11/20/92
3080 * WE WILL NOT INDICATE RE-WRITES BY AN 'R'
3081 * OR AN 'RR' AFTER THE POLICY NUMBER.
3082 *
3083 IF WS-RVHS-GB-POLY-EFF-YYYYMMDD < '19921119'
3084 PERFORM 4015-GET-RR-COUNT
3085 ELSE
3086 NEXT SENTENCE.
3087 *
3088 IF WS-RVHS-GB-POL-PREFIX(1:2) = 'WC'
3089 NEXT SENTENCE
3090 ELSE
3091 PERFORM 4030-GET-CORRECT-PREFIX THRU 4030-EXIT.
3092 *
3093 STRING WS-RVHS-GB-POL-PREFIX DELIMITED BY SPACES,
3094 "-" DELIMITED BY SIZE,
3095 WS-RVHS-POL-NO DELIMITED BY SPACES,
3096 WS-RVHS-GB-POLY-EFF-DATE(10:2) DELIMITED BY SIZE,
3097 WS-REWRITE DELIMITED BY SPACES
3098 INTO WS-NCCI-POLICY-NUMBER.
3099 *
3100 4010-EXIT.
3101 EXIT.
3102 *
3103 4015-GET-RR-COUNT.
3104 *
3105 *
3106 EXEC SQL AT ORA1
3107 SELECT COUNT(*)

```



```

3108 INTO :WS-REWRITE-COUNT
3109 FROM INFC
3110 WHERE INFC_HSTP_GB_CODE_OF = 'RW'
3111 AND INFC_POLY_GB_NO_PART_OF IN
3112 (SELECT POLY_GB_NO
3113 FROM POLY
3114 WHERE POLY_GB_EFF_DATE = :WS-RVHS-GB-POLY-EFF-DATE
3115 AND POLY_GB_POL_NUMBER = RTRIM(:WS-RVHS-POL-NO))
3116 END-EXEC.
3117 *
3118 IF SQLCODE = 0
3119 IF WS-REWRITE-COUNT > 1
3120 MOVE 'RR' TO WS-REWRITE
3121 ELSE
3122 IF WS-REWRITE-COUNT = 1
3123 MOVE 'R ' TO WS-REWRITE
3124 ELSE
3125 MOVE ' ' TO WS-REWRITE
3126 ELSE
3127 IF SQLCODE = 1403
3128 MOVE ' ' TO WS-REWRITE
3129 ELSE
3130 MOVE "*** ERROR ** 4015 COUNTING REWRITES ***"
3131 TO WS-ERR-MSG
3132 PERFORM 9900-ABORT THRU 9900-EXIT.
3133 *
3134 *
3135 4015-EXIT.
3136 EXIT.
3137 4020-GET-EXIT.
3138 *
3139 INITIALIZE WS-EXIT-WC-CLASS-CODE.
3140 *
3141 EXEC SQL AT ORA1
3142 SELECT
3143 EXIT_WC_CLASS_CODE
3144 INTO
3145 :WS-EXIT-WC-CLASS-CODE
3146 FROM EXIT
3147 WHERE EXIT_GB_NO =
3148 (SELECT RSRV_EXIT_GB_NO FROM RSRV
3149 WHERE RSRV_CL_NO = :WS-RVHS-CL-NO)
3150 END-EXEC.
3151 *
3152 IF SQLCODE NOT EQUAL 0 AND SQLCODE NOT EQUAL 1403
3153 MOVE "*** ERROR ** SR-4020 GETTING EXIT" TO WS-ERR-MSG
3154 PERFORM 9900-ABORT THRU 9900-EXIT.
3155 *
3156 4020-EXIT.
3157 EXIT.
3158 *
3159 4030-GET-CORRECT-PREFIX.
3160 *
3161 INITIALIZE WS-PLSM-GB-POL-PREFIX.
3162 *
3163 EXEC SQL AT ORA1

```

```

3164         SELECT PLSM_GB_POL_PREFIX
3165         INTO :WS-PLSM-GB-POL-PREFIX
3166         FROM PLSM
3167         WHERE PLSM_GB_NO = :WS-RVHS-PLSM-GB-NO
3168     END-EXEC.
3169 *
3170     IF SQLCODE = 0
3171         NEXT SENTENCE
3172     ELSE
3173         MOVE '** ERROR GETTING PLSM_GB_NO IN 4030' TO WS-ERR-MSG
3174         DISPLAY 'RVHS_PLSM_GB_NO =' WS-RVHS-PLSM-GB-NO
3175         PERFORM 9900-ABORT THRU 9900-EXIT.
3176 *
3177     MOVE WS-PLSM-GB-POL-PREFIX TO WS-RVHS-GB-POL-PREFIX.
3178 *
3179     4030-EXIT.
3180     EXIT.
3181 *
3182 *
3183     4090-GET-LSPD-STUFF.
3184 *
3185 *
3186     INITIALIZE WS-LSPD-EXP-PAYMENT-AMT.
3187 *
3188     EXEC SQL AT ORA1
3189         SELECT ROUND(SUM(LSPD_CL_PAID_AMOUNT))
3190         INTO :WS-LSPD-EXP-PAYMENT-AMT
3191         FROM LSPD
3192         WHERE LSPD_RVHS_CL_NO IN
3193             (SELECT RVHS_CL_NO
3194              FROM RVHS
3195              WHERE RVHS_CL_CLAIMN_NO = :WS-RVHS-CL-CLAIM-NO)
3196         AND LSPD_DRTP_GB_TYPE IN ('CKEP','CDEP','CDPE')
3197         AND LSPD_CL_PAID_DATE
3198             <= TO_DATE(:WS-RERUN-DATE,'DD-MON-YYYY')
3199     END-EXEC.
3200 *
3201     IF SQLCODE NOT EQUAL 0 AND SQLCODE NOT EQUAL 1403
3202         MOVE '** ERROR ** SR-4090 LSPD2" TO WS-ERR-MSG
3203         PERFORM 9900-ABORT THRU 9900-EXIT.
3204 *
3205     4090-EXIT.
3206     EXIT.
3207 *-----*
3208 *              5000-END-REPORT-TOTALS              *
3209 *-----*
3210 *
3211     5000-END-REPORT-TOTALS.
3212
3213
3214 *
3215     WRITE PRINT-LINE FROM P-HEAD-12 AFTER ADVANCING 1 LINE.
3216 *
3217     MOVE WS-ALL-CLAIMS TO P-TOTAL-CLAIMS.
3218     MOVE WS-INDEM-TOTAL TO P-TOTAL-INDEM.
3219     MOVE WS-MED-TOTAL TO P-TOTAL-MED.

```

```

3220 *
3221     MOVE WS-ALL-CUR-RESV TO P-TOTAL-RESERVE.
3222     MOVE WS-ALL-PAID    TO P-TOTAL-PAID.
3223     MOVE WS-ALL-DEDUCT  TO P-TOTAL-DEDUCT.
3224 *
3225     MOVE "N" TO SW-PROCESS-STATE-RPT2.
3226     PERFORM 6000-PRINT-TOTALS THRU 6000-EXIT.
3227 *
3228     WRITE PRINT-LINE FROM P-HEAD-11 AFTER ADVANCING 1 LINE.
3229 *
3230     MOVE 0 TO WS-ALL-CLAIMS,
3231           WS-INDEM-TOTAL,
3232           WS-MED-TOTAL,
3233           WS-ALL-CUR-RESV,
3234           WS-ALL-PAID,
3235           WS-ALL-DEDUCT.
3236 *
3237     5000-EXIT.
3238     EXIT.
3239 *
3240 * -----*
3241 *           6000-PRINT-TOTALS                      *
3242 * -----*
3243 *
3244     6000-PRINT-TOTALS.
3245 *
3246 *
3247     IF SW-PROCESS-STATE-RPT2 = "Y" THEN
3248         WRITE PRINT-LINE2 FROM P-HEAD-7 AFTER ADVANCING 2 LINES
3249         WRITE PRINT-LINE2 FROM P-HEAD-8 AFTER ADVANCING 2 LINES
3250         WRITE PRINT-LINE2 FROM P-HEAD-9 AFTER ADVANCING 1 LINE
3251         WRITE PRINT-LINE2 FROM P-HEAD-10 AFTER ADVANCING 2 LINES
3252         MOVE SPACES TO PRINT-LINE2.
3253 *
3254     WRITE PRINT-LINE FROM P-HEAD-7 AFTER ADVANCING 2 LINES.
3255     WRITE PRINT-LINE FROM P-HEAD-8 AFTER ADVANCING 2 LINES.
3256     WRITE PRINT-LINE FROM P-HEAD-9 AFTER ADVANCING 1 LINE.
3257     WRITE PRINT-LINE FROM P-HEAD-10 AFTER ADVANCING 2 LINES.
3258     MOVE SPACES TO PRINT-LINE.
3259 *
3260     6000-EXIT.
3261     EXIT.
3262 *
3263 * -----*
3264 *           7000-PRINT-HEADING                      *
3265 * -----*
3266 *
3267     7000-PRINT-HEADING.
3268 *
3269     ADD 1 TO WS-PAGE.
3270     MOVE WS-PAGE TO P-PGCOUNT,
3271           P-PGCOUNT-ASEQ.
3272 *
3273     MOVE WS-RVHS-GB-STATE TO P-STATE1.
3274 *
3275     IF WS-RVHS-BSOR-GB-FM-ACCT NOT = WS-HOLD-BSOR-GB-FM-ACCT

```

```

3276         IF WS-HOLD-BSOR-GB-FM-ACCT NOT = 0
3277             PERFORM 5000-END-REPORT-TOTALS THRU 5000-EXIT
3278         IF WS-RVHS-RSTP-CL-TYPE-CODE = "WM"
3279             ADD 1 TO WS-MED-TOTAL
3280         ELSE
3281             ADD 1 TO WS-INDEM-TOTAL
3282         END-IF
3283         ADD 1 TO WS-ALL-CLAIMS
3284     END-IF
3285     PERFORM 7100-GET-COMPANY-NAME THRU 7100-EXIT
3286     MOVE WS-BSOR-GB-NAME TO P-COMPANY-NAME,
3287         P-COMPANY-NAME-ASEQ
3288     MOVE WS-RVHS-BSOR-GB-FM-ACCT TO WS-HOLD-BSOR-GB-FM-ACCT
3289 END-IF.
3290 *
3291     IF WS-RERUN = 'Y'
3292         IF WS-UPDATE-SWITCH = 'Y'
3293             WRITE PRINT-LINE FROM P-HEAD-ASEQ AFTER ADVANCING PAGE
3294         END-IF
3295     ELSE
3296         WRITE PRINT-LINE FROM P-HEAD-1 AFTER ADVANCING PAGE
3297     END-IF.
3298
3299     WRITE PRINT-LINE FROM P-HEAD-2 AFTER ADVANCING 1 LINE.
3300     WRITE PRINT-LINE FROM P-HEAD-3 AFTER ADVANCING 1 LINE.
3301     WRITE PRINT-LINE FROM P-HEAD-4 AFTER ADVANCING 4 LINES.
3302     WRITE PRINT-LINE FROM P-HEAD-5 AFTER ADVANCING 1 LINE.
3303     WRITE PRINT-LINE FROM P-HEAD-6 AFTER ADVANCING 1 LINE.
3304 *
3305     MOVE 11 TO WS-LINE-COUNT.
3306     MOVE SPACES TO PRINT-LINE.
3307 *
3308     IF SW-PROCESS-STATE-RPT2 = "Y" THEN
3309         WRITE PRINT-LINE2 FROM P-HEAD-1 AFTER ADVANCING PAGE
3310         WRITE PRINT-LINE2 FROM P-HEAD-2 AFTER ADVANCING 1 LINE
3311         WRITE PRINT-LINE2 FROM P-HEAD-3 AFTER ADVANCING 1 LINE
3312         WRITE PRINT-LINE2 FROM P-HEAD-4 AFTER ADVANCING 4 LINES
3313         WRITE PRINT-LINE2 FROM P-HEAD-5 AFTER ADVANCING 1 LINE
3314         WRITE PRINT-LINE2 FROM P-HEAD-6 AFTER ADVANCING 1 LINE
3315         MOVE SPACES TO PRINT-LINE2.
3316 *
3317     7000-EXIT.
3318     EXIT.
3319 *
3320 * -----*
3321 *           7100-GET-COMPANY-NAME           *
3322 * -----*
3323 *
3324     7100-GET-COMPANY-NAME.
3325 *
3326     INITIALIZE WS-BSOR-GB-NAME.
3327 *
3328     EXEC SQL AT ORA1
3329         SELECT
3330             UPPER(LPAD(BSOR_GB_NAME,((60 - LENGTH(BSOR_GB_NAME))/2) +
3331                 LENGTH(BSOR_GB_NAME),' '))

```

```

3332         INTO :WS-BSOR-GB-NAME
3333         FROM BUSORG
3334         WHERE BSOR_GB_ACCOUNT = :WS-RVHS-BSOR-GB-FM-ACCT
3335     END-EXEC.
3336 *
3337     IF SQLCODE NOT = 0 AND NOT = 1403
3338         MOVE "*** ERROR ** ON SELECT FOR COMPANY NAME - 7100"
3339         TO WS-ERR-MSG
3340         PERFORM 9900-ABORT THRU 9900-EXIT.
3341 *
3342     IF SQLCODE = 1403
3343         MOVE
3344         '                COMPANY NAME NOT FOUND                '
3345         TO WS-BSOR-GB-NAME.
3346 *
3347     7100-EXIT.
3348     EXIT.
3349 *
3350 *-----*
3351 *                7400-CHECK-STATE                                *
3352 *-----*
3353 *
3354     7400-CHECK-STATE.
3355 *
3356         IF WS-HOLD-STATE = STATE1 OR WS-HOLD-STATE = STATE2 OR
3357         WS-HOLD-STATE = STATE3 OR WS-HOLD-STATE = STATE4 OR
3358         WS-HOLD-STATE = STATE5 OR WS-HOLD-STATE = STATE6 THEN
3359             MOVE "Y" TO SW-PROCESS-STATE-RPT2
3360         ELSE
3361             MOVE "N" TO SW-PROCESS-STATE-RPT2
3362         END-IF.
3363 *
3364         IF WS-RVHS-GB-STATE = STATE1 OR WS-RVHS-GB-STATE = STATE2 OR
3365         WS-RVHS-GB-STATE = STATE3 OR WS-RVHS-GB-STATE = STATE4 OR
3366         WS-RVHS-GB-STATE = STATE5 OR WS-RVHS-GB-STATE = STATE6
3367         THEN
3368             MOVE "Y" TO SW-PROCESS-NEXT-STATE
3369         ELSE
3370             MOVE "N" TO SW-PROCESS-NEXT-STATE
3371         END-IF.
3372 *
3373     7400-EXIT.
3374     EXIT.
3375 *
3376 *-----*
3377 *                7500-UPDATE-RESV-HIST                            *
3378 *-----*
3379 *
3380     7500-UPDATE-RESV-HIST.
3381 *
3382         EXEC SQL AT ORA1
3383         UPDATE RVHS
3384             SET RVHS_CL_UNIT_STATS_DATE
3385             = :WS-END-DATE
3386         WHERE RVHS_CL_NO = :WS-RVHS-CL-NO
3387     END-EXEC.

```

```

3388
3389 *
3390     IF SQLCODE NOT = 0
3391         MOVE "*** ERROR ** ON UPDATE RESV_HIST 7500"
3392             TO WS-ERR-MSG
3393         PERFORM 9900-ABORT THRU 9900-EXIT.
3394 *
3395     EXEC SQL AT ORA1
3396     COMMIT WORK
3397     END-EXEC.
3398 *
3399     IF SQLCODE NOT = 0
3400         MOVE "*** ERROR ** ON COMMIT 7500" TO WS-ERR-MSG
3401         PERFORM 9900-ABORT THRU 9900-EXIT.
3402 *
3403     7500-EXIT.
3404     EXIT.
3405 *
3406 *
3407 * -----*
3408 *             8000-GET-UNLS-DATA *
3409 * -----*
3410     8000-GET-UNLS-DATA.
3411 *
3412     MOVE 0     TO WS-UNLS-UNPL-WC-NO.
3413
3414     MOVE 1     TO WS-UNLS-NUM-OF-CLMS.
3415
3416 *****--  MOVE SPACES TO WS-UNLS-LOSS-ACT,
3417 *****--           WS-UNLS-LOSS-TYPE.
3418 *   MOVE 61     TO WS-UNLS-COVG-CODE.
3419 *
3420 * LOSS CONDITIONS - ACT
3421 * WC_COVERAGE_CODE
3422 * WCCC_CL_ACT_TYPE --- S = STATE ACT -- CODE IS 01 FOR UNIT STATS
3423 *           U = USL+HW "F" (LONGSHORE/HARBORS WC) -- CODE IS 02 FOR UNIT
STATS
3424 *           OTHER 3 DO NOT PERTAIN TO US
3425 *
3426     if (WS-UNLS-LOSS-ACT = spaces or
3427         WS-UNLS-LOSS-TYPE = spaces)
3428     EXEC SQL AT ORA1
3429     SELECT DECODE (WCCC_CL_ACT_TYPE,'S','01','U','02',NULL,'01'),
3430             DECODE(SUBSTR(WCCC_CL_TYPE_DESC,1,6),
3431                 'Trauma','01','Occupa','02','Cumula','03',NULL,'00')
3432     INTO   :WS-UNLS-LOSS-ACT,
3433           :WS-UNLS-LOSS-TYPE
3434     FROM   WCCC, RVHS
3435     WHERE  RVHS_WCCC_CL_TYPE_CODE = WCCC_CL_TYPE_CODE
3436     AND    RVHS_WCCC_CL_TYPE_CODE = :WS-RVHS-WCCC-TYPE-CODE-N
3437     AND    RVHS_CL_NO = :WS-RVHS-CL-NO
3438     END-EXEC
3439     END-IF.
3440 *
3441 * CHECK THIS LATER....
3442 *

```



```

3443         IF SQLCODE = 0
3444             NEXT SENTENCE
3445         ELSE
3446             IF SQLCODE = 1403
3447                 MOVE '01' TO WS-UNLS-LOSS-ACT, WS-UNLS-LOSS-TYPE
3448             *
3449             * CHECK THIS ABOVE LATER....
3450             *
3451             ELSE
3452                 MOVE "***ERROR ** GETTING UNLS NEXTVAL PARA-8000 "
3453                     TO WS-ERR-MSG
3454                 PERFORM 9900-ABORT THRU 9900-EXIT
3455             END-IF.
3456
3457
3458             *
3459             *
3460             * LOSS CONDITIONS - TYPE OF COVERAGE
3461             * EXPOSURE TYPES TIED TO THE CLAIM
3462             *
3463             * 1.   WORKER'S COMP ONLY (COVERAGE A EXTP 900 W/O COVERAGE B CLASS
CODES) 01  LOSSES INCURRED UNDER PART I
3464             *
3465             * 2.   EMPLOYER'S LIABILITY ONLY (COVERAGE B 900 W/ COVERAGE B CLASS
CODES) 02  LOSSES INCURRED UNDER PART II
3466             * 3.   BOTH -- LOSS IS TIED TO BOTH                                03  LOSSES INCURRED
UNDER PART I & II
3467             *
3468             * IF A POLICY HAS NO COVERAGE B CLASS CODES THEN LOSSES CAN ONLY BE
INCURRED UNDER PROVISIONS OF PART I (COV A)
3469             *
3470             * IF A POLICY HAS COVERAGE B ONLY (IS THIS POSSIBLE?) THEN LOSSES CAN ONLY
BE INCURRED UNDER PROVISIONS OF PART II (COV B)
3471             * --- HOW CAN WE TELL THAT A POLICY ONLY HAS COVERAGE B ?????? ----
3472             *
3473             * IF A POLICY HAS BOTH (MOST LIKELY I THINK) THEN LOSSES CAN BE INCURRED
UNDER PROVISIONS OF BOTH PART I & II (COV A & B)
3474             *
3475             * ANSWER - WE ONLY WRITE POLICIES FOR BOTH. CODE WILL ALWAYS BE 03
3476             *
3477             *
3478             *
3479             EXEC SQL AT ORA1
3480                 SELECT UNIT_STATS_LOSS_SEQUENCE.NEXTVAL
3481                 INTO :WS-UNLS-CL-NO
3482                 FROM DUAL
3483             END-EXEC.
3484             *
3485             IF SQLCODE NOT EQUAL 0 THEN
3486                 MOVE "***ERROR ** GETTING UNLS NEXTVAL PARA-8000 "
3487                     TO WS-ERR-MSG
3488                 PERFORM 9900-ABORT THRU 9900-EXIT
3489             END-IF.
3490             *
3491             EXEC SQL AT ORA1
3492                 SELECT STAT_GB_STATE_NO

```

```

3493         INTO :WS-RVHS-GB-STATE-NO
3494         FROM STATE
3495         WHERE STAT_GB_ALPHA_STATE
3496         = RTRIM(:WS-RVHS-GB-STATE)
3497     END-EXEC.
3498 *
3499     IF SQLCODE NOT EQUAL 0 THEN
3500         MOVE "***ERROR ** GETTING STATE-NO PARA-8000 "
3501             TO WS-ERR-MSG
3502         PERFORM 9900-ABORT THRU 9900-EXIT
3503     END-IF.
3504 *
3505     EXEC SQL AT ORA1
3506         SELECT UNPL_WC_NO
3507         INTO :WS-UNLS-UNPL-WC-NO
3508         FROM UNPL
3509         WHERE UNPL_WC_POLY_NUMBER = RTRIM(:WS-RVHS-POL-NO)
3510         AND UNPL_WC_POLY_EFF_DATE
3511             = TO_DATE(:WS-RVHS-GB-POLY-EFF-MMDDYYYY,'MMDDYYYY')
3512         AND UNPL_WC_CORRECTION_NO = 0
3513     END-EXEC.
3514 *
3515 * CINDY/ROY THIS NEEDS TO BE CHANGED TO BLOW UP ON 1403.
3516 *
3517     IF SQLCODE EQUAL ZERO
3518         NEXT SENTENCE
3519     ELSE
3520
3521         * WE'VE PICKED UP A CLAIM THAT HAS NO CURRENT POLICY
3522         * BUT DOES HAVE AN OLD ONE AND WE STILL HAVE TO REPORT IT.
3523         * MAKE A SKELETON UNPL FOR IT
3524
3525         IF SQLCODE EQUAL 1403
3526
3527         *****-- PLI32502.9352----->|
3528         *****-- |
3529         *****-- this is dead logic; we now report to NCCI and Independent Bureaus. |
3530         *****-- this logic prevents proper ties between losses and policies |
3531         *****-- multiple codes (sqlcode 1422) now exist for 20 states, incl 15 NCCI|
3532         *****-- |
3533         *****--* COULD BE A STATE THAT IS NOT CURRENTLY REPORTED THRU NCCI |
3534         *****--* DON'T NEED THOSE CLAIMS IN OUR TABLES, AND NOT GETTING |
3535         *****--* PICKED UP THRU WCPOL_STATS |
3536         *****--* NEED TO UPDATE MCTB TABLES.... DE, MA, AND WI ARE INDEPENDENT |
3537         *****-- EXEC SQL AT ORA1 |
3538         *****-- SELECT MCTB_GB_CODE_TYPE |
3539         *****-- INTO :WS-MCTB-GB-CODE-TYPE |
3540         *****-- FROM MCTB |
3541         *****-- WHERE MCTB_GB_CODE = :WS-RVHS-GB-STATE |
3542         *****-- AND MCTB_GB_CODE != 'NJ' |
3543         *****-- OR :WS-RVHS-GB-STATE IN ('DE','MA','WI') |
3544         *****-- END-EXEC |
3545         *****-- IF SQLCODE EQUAL 1403 |
3546         *****-- ----->|
3547
3548         DISPLAY "MAKING A SKELETON"

```

```

3549 EXEC SQL AT ORA1
3550 SELECT UNIT_STATS_POLICY_SEQUENCE.NEXTVAL
3551 INTO :WS-UNLS-UNPL-WC-NO
3552 FROM DUAL
3553 END-EXEC
3554 PERFORM 8050-PRINT-EXCEPTION-REPORT THRU 8050-EXIT
3555 PERFORM 8025-CREATE-SKELETON THRU 8025-EXIT.
3556 *
3557 COMPUTE WS-UNLS-INDEMNITY-AMT ROUNDED =
3558 (CLAIM-INDEMNITY-RESERVE-AMT + CLAIM-INDEMNITY-PAYMENT-AMT).
3559 *
3560 COMPUTE WS-UNLS-MEDICAL-AMT ROUNDED =
3561 (CLAIM-MEDICAL-RESERVE-AMT + CLAIM-MEDICAL-PAYMENT-AMT).
3562
3563 IF WS-UNLS-INDEMNITY-AMT > 0 THEN
3564 MOVE WS-UNLS-INDEMNITY-AMT TO WS-UNLS-CL-CA-GROSS-INCUR
3565 ELSE
3566 IF WS-UNLS-MEDICAL-AMT > 0 THEN
3567 MOVE WS-UNLS-MEDICAL-AMT TO WS-UNLS-CL-CA-GROSS-INCUR
3568 ELSE
3569 MOVE 0 TO WS-UNLS-CL-CA-GROSS-INCUR
3570 END-IF.
3571
3572 *****
3573 * INCURRED VOC REHAB: PAID IS FOUND FROM LOOKING AT DRAFTS/WHOM
3574 WRITTEN TO ETC
3575 * (SEE 1800-)
3576 *
3577 IF WS-UNLS-CLM-STATUS = 'O'
3578 IF WS-RSRV-CL-REHAB > WS-UNLS-CL-PAID-VOC-REHAB
3579 MOVE WS-RSRV-CL-REHAB TO WS-UNLS-CL-INCUR-VOC-REHAB
3580 ELSE
3581 MOVE WS-UNLS-CL-PAID-VOC-REHAB TO WS-UNLS-CL-INCUR-VOC-REHAB
3582 END-IF
3583 ELSE
3584 MOVE WS-UNLS-CL-PAID-VOC-REHAB TO WS-UNLS-CL-INCUR-VOC-REHAB
3585 END-IF.
3586 **
3587 IF WS-UNLS-CL-INCUR-VOC-REHAB NOT EQUAL 0
3588 MOVE 'Y' TO WS-UNLS-CLM-VOC-REHAB-IND
3589 ELSE
3590 MOVE 'N' TO WS-UNLS-CLM-VOC-REHAB-IND
3591 END-IF.
3592 *****
3593 * SUBROGATION VALUE HERE IS CLAIM TOTAL W/O SUBROGATION CREDIT - subro is
3594 negative
3595 *
3596 IF WS-SUBRO NOT EQUAL ZERO
3597 COMPUTE WS-SUBRO-PART ROUNDED = WS-UNLS-INDEMNITY-AMT
3598 + WS-UNLS-MEDICAL-AMT
3599 - WS-SUBRO
3600 ELSE
3601 MOVE ZEROES TO WS-SUBRO-PART.
3602 *****
3603 * :WHERE THE INJURY WAS RULED AS CAUSED OVER A PERIOD OF TIME,

```

```

3603 *           WITH 2 DIFFERENT INSURERS PROVIDING COVERAGE DURING THE PERIOD.
3604 *           THIS IS THE PERCENT OF THE CLAIM WE SHOULD END UP PAYING.
3605 *           A CLOSED CLAIM WILL HAVE RECORDED PAYMENT FOR THE OTHER
COMPANY'S
3606 *           PERCENTAGE, SO WE DON'T APPLY PERCENTAGE. A REOPENED CLAIM
3607 *           SHOULD PROBABLY BE LOOKED AT - IF SOME MONEY HAS COME IN AND
3608 *           THE OTHER COMPANY OWES MORE, THE PERCENTAGE WILL NOT WORK.
3609 *           FYI- MONEY RECOVERED IS CODED LIKE ANY SUBROGATION PAYMENT
3610 *
3611 IF WS-CLOC-CL-JOINT-COV-PERCENT NOT EQUAL ZERO
3612 IF WS-UNLS-CLM-STATUS = 'O'
3613     COMPUTE WS-JOINT ROUNDED = (WS-UNLS-INDEMNITY-AMT
3614     + WS-UNLS-MEDICAL-AMT)
3615     * (WS-CLOC-CL-JOINT-COV-PERCENT / 100)
3616 ELSE
3617     COMPUTE WS-JOINT = WS-UNLS-INDEMNITY-AMT
3618     + WS-UNLS-MEDICAL-AMT
3619 ELSE
3620     MOVE ZEROES TO WS-JOINT.
3621 *****
3622 * S-CLAIMS : AMOUNT ENTERED IN CLENTY1 FORM
3623 *
3624 IF WS-CLOC-CL-S-CLAIM NOT EQUAL ZERO
3625     MOVE WS-CLOC-CL-S-CLAIM TO WS-S-CLAIM
3626 ELSE
3627     MOVE ZEROES TO WS-S-CLAIM.
3628 *****
3629 * TOTAL
3630 COMPUTE WS-UNLS-CL-CA-GROSS-INCUR =
3631     WS-SUBRO-PART + WS-JOINT + WS-S-CLAIM.
3632 *
3633 *
3634     MOVE '00' TO WS-UNLS-CLM-HMO-TYPE;
3635
3636 if WS-RVHS-GB-STATE-NO not equal '04'
3637 *****--PLI32502.11843
3638     EXEC SQL AT ORA1
3639         SELECT DECODE(UNPL_WC_MANG_CARE_ORG_IND,'Y','01','00')
3640         INTO :WS-UNLS-CLM-HMO-TYPE
3641         FROM UNIT_STATS_POLICY
3642         WHERE UNPL_WC_NO = :WS-UNLS-UNPL-WC-NO
3643     END-EXEC
3644 *****--
3645     end-if.
3646
3647     MOVE WS-UNLS-LOSS-ACT TO WS-UNLS-COVG-CODE;
3648
3649 *
3650 * HAVE TO CHECK JURISDICTION STATE -- MAG TAP MANUAL
3651 * SAYS ENTER STATE CODE FOR JURISDICTION STATE WHEN
3652 * A DIFFERENT STATE IS GOING TO ADMINISTER THE CLAIM
3653 * AND WHOSE STATUTES APPLY, THEN THE EXPOSURE STATE.
3654 * CHANGED USRD JURISDICTION STATE TO BE 00 FOR NOW...
3655 *
3656 *
3657 * MAY HAVE TO DECODE WS-UNLS-CLM-CSE-OF-INJ-CDE

```

```

3658 * (WS-RVHS-COLT-CL-TYPE-CODE) 4 CHARS, LETTERS THAT
3659 * DON'T CORRESPOND TO UNIT STATS SPECIFIED CODES
3660 * SUBSTRINGING OTHERS IE, NOSI AND POBI IN USRDINP.
3661 * MAY NOT HAVE ENOUGH ROOM.... TO DECODE THIS ONE
3662 * IN FORM.
3663 *
3664 EVALUATE WS-RVHS-COLT-CL-TYPE-CODE
3665     WHEN "HITO"
3666         MOVE '68' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3667     WHEN "ABFE"
3668         MOVE '74' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3669     WHEN "ABUP"
3670         MOVE '81' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3671     WHEN "BITE"
3672         MOVE '85' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3673     WHEN "BURN"
3674         MOVE '84' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3675     WHEN "CUTW"
3676         MOVE '19' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3677     WHEN "DERM"
3678         MOVE '09' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3679     WHEN "FALL"
3680         MOVE '31' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3681     WHEN "FUME"
3682         MOVE '82' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3683     WHEN "HART"
3684         MOVE '60' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3685     WHEN "HITB"
3686         MOVE '75' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3687     WHEN "PNCW"
3688         MOVE '19' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3689     WHEN "TWLB"
3690         MOVE '53' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3691     WHEN "VEHC"
3692         MOVE '50' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3693     WHEN "INTI"
3694         MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3695     WHEN "AGGR"
3696         MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3697     WHEN "ENVI"
3698         MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3699     WHEN "ORGN"
3700         MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3701     WHEN "OTHR"
3702         MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3703     WHEN "PLAY"
3704         MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3705     WHEN "UNDT"
3706         MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3707     WHEN "GUNS"
3708         MOVE '93' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3709     WHEN OTHER
3710         DISPLAY 'WS-RVHS-COLT-CL-TYPE-CODE ';
3711             WS-RVHS-COLT-CL-TYPE-CODE
3712         MOVE
3713 "ERROR EVALUATING WS-RVHS-COLT-CL-TYPE-CODE IN 8000-"

```

```

3714             TO WS-ERR-MSG
3715             MOVE '99' TO WS-UNLS-CLM-CSE-OF-INJ-CDE
3716 *****      PERFORM 9900-ABORT THRU 9900-EXIT
3717             END-EVALUATE.
3718             PERFORM 8100-INSERT-UNLS-DATA THRU 8100-EXIT.
3719 *
3720             8000-EXIT.
3721             EXIT.
3722 *
3723             8025-CREATE-SKELETON.
3724 *
3725             MOVE 'N' TO WS-UNPL-XMIT-IND.
3726
3727             EXEC SQL AT ORA1
3728             INSERT INTO UNPL (UNPL_WC_NO,
3729                             UNPL_WC_REPORT_DATE,
3730                             UNPL_WC_NCCI_POL_NUMBER,
3731                             UNPL_WC_POLY_NUMBER,
3732                             UNPL_WC_POLY_EFF_DATE,
3733                             UNPL_STAT_GB_STATE_NO,
3734                             UNPL_WC_CORRECTION_NO,
3735                             UNPL_WC_POLY_EXP_CAN_DATE,
3736                             UNPL_UNIT_XMIT)
3737             VALUES (:WS-UNLS-UNPL-WC-NO,
3738                     :WS-UNLS-REPT-DATE,
3739                     :WS-RVHS-GB-POL-PREFIX||:WS-RVHS-POL-NO,
3740                     :WS-RVHS-POL-NO,
3741                     :WS-RVHS-GB-POLY-EFF-DATE,
3742                     :WS-RVHS-GB-STATE-NO,
3743                     0,
3744                     :WS-RVHS-GB-POLY-EXP-DATE,
3745                     :WS-UNPL-XMIT-IND)
3746             END-EXEC.
3747 *
3748             DISPLAY "BEGIN".
3749             DISPLAY "WS-UNLS-UNPL-WC-NO " WS-UNLS-UNPL-WC-NO.
3750             DISPLAY "WS-END-DATE "      WS-END-DATE.
3751             DISPLAY "PREFIX "           WS-RVHS-GB-POL-PREFIX.
3752             DISPLAY "WS-RVHS-POL-NO "   WS-RVHS-POL-NO.
3753             DISPLAY "WS-RVHS-GB-POLY-EFF-DATE "
3754                     WS-RVHS-GB-POLY-EFF-DATE.
3755             DISPLAY "WS-RVHS-GB-STATE-NO "
3756                     WS-RVHS-GB-STATE-NO.
3757             DISPLAY "WS-RVHS-GB-POLY-EXP-DATE "
3758                     WS-RVHS-GB-POLY-EXP-DATE.
3759             DISPLAY "END".
3760 *             PERFORM 9900-ABORT THRU 9900-EXIT.
3761 *
3762             IF SQLCODE NOT EQUAL 0 THEN
3763             DISPLAY "ERROR TRYING TO INSERT INTO UNPL IN 8025-PARA"
3764             DISPLAY "SQLCODE IS "      SQLCODE
3765             DISPLAY "WS-UNLS-UNPL-WC-NO " WS-UNLS-UNPL-WC-NO
3766             DISPLAY "WS-END-DATE "      WS-END-DATE
3767             DISPLAY "PREFIX "           WS-RVHS-GB-POL-PREFIX
3768             DISPLAY "WS-RVHS-POL-NO "   WS-RVHS-POL-NO
3769             DISPLAY "WS-RVHS-GB-POLY-EFF-DATE "

```



```

3770             WS-RVHS-GB-POLY-EFF-DATE
3771     DISPLAY "WS-RVHS-GB-STATE-NO "
3772             WS-RVHS-GB-STATE-NO
3773     DISPLAY "WS-RVHS-GB-POLY-EXP-DATE "
3774             WS-RVHS-GB-POLY-EXP-DATE
3775 *     PERFORM 9900-ABORT THRU 9900-EXIT
3776     ELSE
3777         PERFORM 9000-COMMIT THRU 9000-EXIT
3778     END-IF.
3779
3780     8025-EXIT.
3781     EXIT.
3782
3783     8050-PRINT-EXCEPTION-REPORT.
3784 *
3785         IF WS-LINE-COUNT2 > 38 THEN
3786             PERFORM 8075-EXCEPTION-HEADINGS THRU 8075-EXIT.
3787 *
3788             MOVE WS-RVHS-WCCC-TYPE-CODE-C TO E-TYPE-CODE.
3789             MOVE WS-RVHS-POL-NO          TO E-POL-NO.
3790             MOVE WS-RVHS-CLASS-CODE      TO E-CLASS-CODE.
3791             MOVE WS-RVHS-GB-STATE        TO E-STATE.
3792             MOVE WS-RVHS-CL-OCC-DATE     TO E-OCC-DATE.
3793             MOVE WS-RVHS-CL-CLAIM-NO     TO E-CLAIM-NO.
3794             MOVE WS-RVHS-RSTP-CL-TYPE-CODE TO E-RESV-TYPE.
3795             MOVE WS-RVHS-CLIC-CL-CODE    TO E-INJURY.
3796             MOVE WS-CURRENT-RESV        TO E-CURR-RESV.
3797             MOVE WS-PAID                 TO E-PAID-AMT.
3798             MOVE WS-RVHS-GB-POLY-EFF-DATE TO E-EFF-DATE.
3799 *
3800             WRITE PRINT-LINE3 FROM EXCEPTION-DETAIL
3801                 AFTER ADVANCING 1 LINE.
3802             MOVE SPACES TO PRINT-LINE3.
3803             ADD +1 TO WS-LINE-COUNT2.
3804 *
3805             8050-EXIT.
3806             EXIT.
3807 *
3808             8075-EXCEPTION-HEADINGS.
3809 *
3810             ADD 1 TO WS-PAGE-COUNT.
3811 *
3812             MOVE WS-PAGE-COUNT TO E-PAGE.
3813             WRITE PRINT-LINE3 FROM E-HEAD-1 AFTER ADVANCING PAGE.
3814             MOVE SPACES TO PRINT-LINE3.
3815 *             WRITE PRINT-LINE3 FROM P-HEAD-1.
3816             MOVE SPACES TO PRINT-LINE3.
3817             WRITE PRINT-LINE3 FROM P-HEAD-2 AFTER 2.
3818             MOVE SPACES TO PRINT-LINE3.
3819             WRITE PRINT-LINE3 FROM E-HEAD-2 AFTER 2.
3820             MOVE SPACES TO PRINT-LINE3.
3821             WRITE PRINT-LINE3 FROM X-HEAD-4 AFTER 2.
3822             MOVE SPACES TO PRINT-LINE3.
3823             WRITE PRINT-LINE3 FROM X-HEAD-5.
3824             MOVE 0 TO WS-LINE-COUNT2.
3825 *

```

```

3826      8075-EXIT.
3827      EXIT.
3828
3829      *-----*
3830      *                8100-INSERT-UNLS-DATA                *
3831      *-----*
3832      8100-INSERT-UNLS-DATA.
3833      *
3834      MOVE 0      TO WS-UNLS-PREV-REP-IND.
3835      MOVE SPACES TO WS-UNLS-UPD-TYPE.
3836      MOVE WS-RPT-SEQ TO WS-UNLS-REPT-NO.
3837      move WS-UNLS-CLM-ALLOC-LAE-PD to WS-UNLS-CLM-ALLOC-LAE-IN.
3838      *****
3839      * FOR NOW, THIS WILL INDICATE A CUSTOM RUN
3840
3841      IF WS-RERUN = 'Y'
3842          IF WS-UPDATE-SWITCH = 'Y'
3843              EXEC SQL AT ORA1
3844                  SELECT DISTINCT CURRENT_SEQ_NO  RPTSEQ
3845                  INTO   :WS-UNLS-REPT-NO-TECH
3846                  FROM TECH
3847                  WHERE CLAIM = :WS-RVHS-CL-CLAIM-NO
3848              END-EXEC
3849          END-IF
3850      END-IF.
3851
3852      IF WS-RERUN = 'Y'
3853          IF WS-UPDATE-SWITCH = 'Y'
3854              EXEC SQL AT ORA1
3855                  SELECT NVL(MAX(UCLA_LAST_REPORT),0) RPTSEQ
3856                  INTO   :WS-UNLS-REPT-NO-UCLA
3857                  FROM UCLA
3858                  WHERE UCLA_CLAIM_NO = :WS-UNLS-CLM-NUMBER
3859              END-EXEC
3860          END-IF
3861      END-IF.
3862
3863      IF WS-UNLS-REPT-NO-UCLA = 0
3864          IF WS-UNLS-REPT-NO-TECH = 1
3865              ADD 1 TO WS-UNLS-REPT-NO
3866          END-IF
3867      ELSE
3868          COMPUTE WS-UNLS-REPT-NO = WS-UNLS-REPT-NO-UCLA + 1
3869      END-IF.
3870
3871      *
3872      *****
3873      *
3874      *****
3875      * PLI32502.16651: GENERATE CATASTROPHE NUMBER.
3876
3877          INITIALIZE WS-CAT-IND,
3878                  WS-RLOC-NO.
3879          MOVE ZERO TO WS-CAT-IND.
3880
3881      * DETERMINE IF THERE IS A CAT NUMBER

```

```

3882
3883
3884     EXEC SQL AT ORA1
3885     SELECT distinct 1 ,
3886     CLOC.CLOC_RLOC_CL_NO
3887     INTO WS-CAT-IND,
3888     WS-RLOC-NO
3889     FROM CLAIM_OCCURRENCE CLOC,
3890     RESV_HIST RVHS
3891     WHERE CLOC.CLOC_CL_NO      = RVHS.RVHS_CLOC_CL_NO
3892     AND CLOC.CLOC_RLOC_CL_NO  IS NOT NULL
3893     AND CLOC.CLOC_CL_CLAIM_TYPE = 'W'
3894     AND RVHS.RVHS_GB_POL_NUMBER = :WS-RVHS-POL-NO
3895     AND CLOC.CLOC_CL_OCCURRENCE_DATE
3896             BETWEEN :WS-BEG-DATE-18 AND
3897             ADD_MONTHS(:WS-BEG-DATE-18,12)
3898     AND CLOC.CLOC_CL_CLAIM_NO = :WS-RVHS-CL-CLAIM-NO
3899     ORDER BY CLOC.CLOC_RLOC_CL_NO
3900     END-EXEC.
3901
3902 ***
3903
3904
3905     IF WS-RLOC-NO NOT = WS-RLOC-HD
3906
3907 * DETERMINE THE CORRECT CAT NUMBER
3908
3909     IF WS-CAT-IND > 0
3910     EXEC SQL AT ORA1
3911     SELECT NVL (MAX (UNLS.UNLS_CL_CAT_NUMBER), 0)
3912             AS MAX_CAT
3913     INTO WS-CAT-NUMBER
3914     FROM UNIT_STATS_LOSS UNLS,
3915     UNIT_STATS_POLICY UNPL
3916     WHERE UNLS.UNLS_UNPL_WC_NO = UNPL.UNPL_WC_NO
3917     AND UNPL.UNPL_WC_NO =
3918     (SELECT MAX (UNPL_WC_NO)
3919     FROM UNPL,
3920     UNLS
3921     WHERE UNLS_UNPL_WC_NO      = UNPL_WC_NO
3922     AND UNPL_WC_POLY_NUMBER    = :WS-RVHS-POL-NO
3923     AND NVL (UNLS_CL_CAT_NUMBER, 0) >= 0)
3924     END-EXEC
3925
3926 ***
3927
3928
3929     COMPUTE WS-CAT-NUMBER = WS-CAT-NUMBER + WS-CAT-IND
3930 *****-- PLI32502.21890      MOVE WS-CAT-NUMBER TO WS-UNLS-CLM-CAT-NO
3931     MOVE 0 TO WS-UNLS-CLM-CAT-NO
3932
3933     IF WS-CAT-NUMBER NOT EQUAL 0
3934     DISPLAY 'CAT: ' WS-UNLS-CLM-CAT-NO
3935     END-IF
3936
3937 *****-- END WS-CAT-IND > 0

```

```

3938         END-IF
3939
3940 *****-- END WS-RLOC-NO NOT = WS-RLOC-HD
3941         END-IF.
3942
3943         MOVE WS-RLOC-NO TO WS-RLOC-HD.
3944
3945
3946 *****
3947 *****--PLI32502.18292
3948         IF WS-LSPD-DFCM-CL-TYPE-CODE-ASWG = 'WC5'
3949             MOVE 'Y' TO WS-UNLS-CLM-LUMP-SUM-IND
3950         END-IF.
3951
3952         INITIALIZE WS-LAWSUIT-IND.
3953
3954         EXEC SQL AT ORA1
3955             SELECT CLOC_CL_LAWSUIT
3956             INTO WS-LAWSUIT-IND
3957             FROM CLAIM_OCCURRENCE CLOC
3958             WHERE CLOC.CLOC_CL_CLAIM_NO = :WS-RVHS-CL-CLAIM-NO
3959         END-EXEC.
3960
3961 *****-- these percentages were dictated by CLAIMS
3962         if WS-RVHS-GB-STATE NOT = 'CA'
3963             IF WS-LAWSUIT-IND = 'Y'
3964                 COMPUTE WS-UNLS-CLM-CLAIMMAINT-ATT-FEE =
3965                     WS-UNLS-INDEMNITY-AMT * .15
3966             *
3967                 COMPUTE WS-UNLS-CLM-EMPLOYER-ATT-FEE =
3968                     WS-UNLS-INDEMNITY-AMT * .25
3969             end-if
3970         END-IF.
3971
3972 *****
3973
3974
3975         EXEC SQL AT ORA1
3976             INSERT INTO UNLS
3977                 (UNLS_CL_NO,
3978                 UNLS_UNPL_WC_NO,
3979                 UNLS_CL_REPORT_DATE,
3980                 UNLS_CL_RVHS_CLASS_CODE,
3981                 UNLS_CL_PREV_REP_IND,
3982                 UNLS_CL_LOSS_COVG_CODE,
3983                 UNLS_CL_NUMBER_OF_CLAIMS,
3984                 UNLS_CL_ACCIDENT_DATE,
3985                 UNLS_CL_CLAIM_NUMBER,
3986                 UNLS_CL_CLAIM_STATUS,
3987                 UNLS_CL_INJURY_CODE,
3988                 UNLS_CL_CAT_NUMBER,
3989                 UNLS_CL_INDEMNITY_AMOUNT,
3990                 UNLS_CL_MEDICAL_AMOUNT,
3991                 UNLS_CL_SSN,
3992                 UNLS_CL_UPDATE_TYPE,
3993                 UNLS_CL_LOSS_ACT,

```

3994	UNLS_CL_LOSS_TYPE,
3995	UNLS_CL_LOSS_RECOVERY,
3996	UNLS_CL_LOSS_COVERAGE,
3997	UNLS_CL_LOSS_SETTLEMENT,
3998	UNLS_STAT_GB_STATE_NO,
3999	UNLS_CL_MANG_CARE_ORG_TYPE,
4000	UNLS_CL_INJ_DESC_CODE_PART,
4001	UNLS_CL_INJ_DESC_CODE_NAT,
4002	UNLS_CL_INJ_DESC_CODE_CLAUSE,
4003	UNLS_CL_OCCUPATION_DESC,
4004	UNLS_CL_VOCATION_REHAB_IND,
4005	UNLS_CL_LUMP_SUM_IND,
4006	UNLS_CL_FRAUD_CLAIM_IND,
4007	UNLS_CL_DEDUCT_CODE,
4008	UNLS_CL_PAID_INDEMNITY_AMT,
4009	UNLS_CL_PAID_MEDICAL_AMT,
4010	UNLS_CL_CLAIMANT_ATT_FEE,
4011	UNLS_CL_EMPLOYER_ATT_FEE,
4012	UNLS_CL_ALAE_PAID_AMT,
4013	UNLS_CL_ALAE_INCURRED_AMT,
4014	UNLS_CL_DEDUCT_AMT,
4015	UNLS_US_HIREDATE,
4016	UNLS_CL_REPORT_NO)
4017	VALUES (:WS-UNLS-CL-NO,
4018	:WS-UNLS-UNPL-WC-NO,
4019	:WS-UNLS-REPT-DATE,
4020	:WS-RVHS-CLASS-CODE,
4021	:WS-UNLS-PREV-REP-IND,
4022	:WS-UNLS-COVG-CODE,
4023	:WS-UNLS-NUM-OF-CLMS,
4024	:WS-RVHS-CL-OCC-DATE,
4025	:WS-UNLS-CLM-NUMBER,
4026	:WS-UNLS-CLM-STATUS,
4027	:WS-UNLS-CLM-INJURY-CODE,
4028	:WS-UNLS-CLM-CAT-NO,
4029	:WS-UNLS-INDEMNITY-AMT,
4030	:WS-UNLS-MEDICAL-AMT,
4031	:WS-UNLS-CLM-SSN,
4032	:WS-UNLS-UPD-TYPE,
4033	:WS-UNLS-LOSS-ACT,
4034	:WS-UNLS-LOSS-TYPE,
4035	:WS-UNLS-LOSS-RECOV,
4036	:WS-UNLS-LOSS-COVG,
4037	:WS-UNLS-LOSS-SETL,
4038	:WS-RVHS-GB-STATE-NO,
4039	:WS-UNLS-CLM-HMO-TYPE,
4040	:WS-UNLS-CLM-BODY-PART-CDE,
4041	:WS-UNLS-CLM-NAT-OF-INJ-CDE,
4042	:WS-UNLS-CLM-CSE-OF-INJ-CDE,
4043	:WS-UNLS-CLM-OCCUP-DESC,
4044	:WS-UNLS-CLM-VOC-REHAB-IND,
4045	:WS-UNLS-CLM-LUMP-SUM-IND,
4046	:WS-UNLS-CLM-FRAUD-IND,
4047	:WS-UNLS-CLM-DEDUCT-CODE,
4048	:WS-UNLS-CLM-IDMNTY-PYMT-AMT,
4049	:WS-UNLS-CLM-MED-PYMT-AMT,

```

4050      :WS-UNLS-CLM-CLAIMAINT-ATT-FEE,
4051      :WS-UNLS-CLM-EMPLOYER-ATT-FEE,
4052      :WS-UNLS-CLM-ALLOC-LAE-PD,
4053      :WS-UNLS-CLM-ALLOC-LAE-IN,
4054      NULL,
4055      :WS_CLPT_CL_HIRE_DATE,
4056      :WS-UNLS-REPT-NO)
4057  END-EXEC.
4058
4059  *
4060      IF SQLCODE NOT EQUAL 0 THEN
4061          MOVE "ERROR TRYING TO INSERT INTO UNLS IN 8100-PARA"
4062              TO WS-ERR-MSG
4063          DISPLAY 'UNPL ID: ' WS-UNLS-UNPL-WC-NO
4064          display 'CLAIM: ' WS-UNLS-CLM-NUMBER
4065          DISPLAY 'REPORT DATE: ' WS-UNLS-REPT-DATE
4066          PERFORM 9900-ABORT THRU 9900-EXIT
4067      ELSE
4068          PERFORM 9000-COMMIT THRU 9000-EXIT
4069          ADD 1 TO WS-UNLS-INSERTS
4070      END-IF.
4071  *****
4072  *****
4073  * UPDATE VALUE OF UNLS_CL_DEDUCT_AMT
4074  *****
4075  *****
4076  *BEGIN PLI31530.3248***
4077      EXEC SQL AT ORA1
4078          UPDATE UNLS
4079              SET UNLS_CL_DEDUCT_AMT      = :WS-UNLS-CL-DEDUCT-AMT
4080              WHERE UNLS_CL_NO           = :WS-UNLS-CL-NO
4081              AND UNLS_UNPL_WC_NO        = :WS-UNLS-UNPL-WC-NO
4082              AND UNLS_CL_REPORT_DATE    = :WS-UNLS-REPT-DATE
4083      END-EXEC.
4084  *END PLI31530.3248***
4085
4086  *****
4087  **** UPDATE OF NEW FIELDS - IF INCLUDED WITH THE ABOVE INSERT THE PROGRAM
4088  ***** HANGS AT THE INSERT. UNDOCUMENTED FEATURE I CAN'T
4089  ***** FIND A SOLUTION FOR. AT LEAST THIS WORKS... DG
4090      EXEC SQL AT ORA1
4091          UPDATE UNLS
4092              SET UNLS_CL_AVG_WKLY_WAGE
4093                  = :WS-UNLS-CL-AVG-WKLY-WAGE,
4094              UNLS_CL_INCUR_VOC_REHAB
4095                  = :WS-UNLS-CL-INCUR-VOC-REHAB,
4096              UNLS_CL_CA_GROSS_INCUR
4097                  = :WS-UNLS-CL-CA-GROSS-INCUR,
4098              UNLS_CL_PERM_DISABL_RATING
4099                  = :WS-CLPT-CL-PERM-DISABL-RATING,
4100              UNLS_CL_REPORT_NO
4101                  = :WS-UNLS-REPT-NO
4102              WHERE UNLS_CL_NO           = :WS-UNLS-CL-NO AND
4103                  UNLS_UNPL_WC_NO        = :WS-UNLS-UNPL-WC-NO
4104                  AND UNLS_CL_REPORT_DATE = :WS-UNLS-REPT-DATE
4105      END-EXEC.

```



```

4106 *
4107     IF SQLCODE NOT EQUAL 0 THEN
4108         MOVE "ERROR TRYING TO UPDATE UNLS IN 8100-PARA"
4109             TO WS-ERR-MSG
4110         PERFORM 9900-ABORT THRU 9900-EXIT
4111     ELSE
4112         PERFORM 9000-COMMIT THRU 9000-EXIT
4113     END-IF.
4114 *
4115     MOVE WS-UNLS-PREV-REP-IND TO HOLD-UNLS-PREV-REP-IND.
4116     MOVE WS-UNLS-UPD-TYPE    TO HOLD-UNLS-UPD-TYPE.
4117     MOVE WS-UNLS-UNPL-WC-NO  TO HOLD-UNLS-UNPL-WC-NO.
4118     MOVE WS-RVHS-CLASS-CODE  TO HOLD-RVHS-CLASS-CODE.
4119     MOVE WS-UNLS-REPT-DATE   TO HOLD-UNLS-REPT-DATE.
4120     MOVE WS-UNLS-CLM-NUMBER  TO HOLD-UNLS-CLM-NUMBER.
4121
4122
4123
4124 *
4125     8100-EXIT.
4126     EXIT.
4127 *-----|9000-COMMIT|-----*
4128 *-----*
4129     9000-COMMIT.
4130 *
4131     EXEC SQL AT ORA1
4132         COMMIT
4133     END-EXEC.
4134 *
4135     9000-EXIT.
4136     EXIT.
4137 *-----*
4138 *             9900-ABORT             *
4139 *-----*
4140     9900-ABORT.
4141 *
4142     MOVE "Y" TO SW-ABORT.
4143 *
4144     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT.
4145 *
4146     9900-EXIT.
4147     EXIT.
4148 *-----|9990-ROLLBACK|-----*
4149 *-----*
4150     9990-ROLLBACK.
4151 *
4152     EXEC SQL AT ORA1
4153         ROLLBACK WORK RELEASE
4154     END-EXEC.
4155 *
4156     9990-EXIT.
4157     EXIT.
4158 *-----|9999-DISPLAY-ERROR|-----*
4159 *-----*
4160     9999-DISPLAY-ERROR.
4161 *

```

```
4162      DISPLAY WS-ERR-MSG.
4163      *
4164      DISPLAY "SQLCODE  = "; SQLCODE .
4165      *
4166      *
4167      IF ABORT
4168          PERFORM 9990-ROLLBACK THRU 9990-EXIT
4169          DISPLAY "ABENDSYM:YES"
4170          STOP RUN.
4171      *
4172      9999-EXIT.
4173      EXIT.
4174      *
4175      *-----*
4176      *          ***  END OF PROGRAM  ***          *
4177      *-----*
4178
```