

IDENTIFICATION DIVISION.
PROGRAM-ID. AUDECPRT.
AUTHOR. ROY WEHLING.
DATE-WRITTEN. 31-JAN-91.

* PROGRAM NAME : AUDECPRT.PCO *

* FUNCTION : THE PURPOSE OF THIS PROGRAM IS TO PRINT THE AUTO
* DEC PAGE FOR ENTRIES IN THE PPRT TABLE WITH A
* PPRT_GB_DONE = 'D' AND PPRT_GB_USER IS EQUAL TO THE
* USER WHO IS RUNNING THIS PROGRAM. *

MODIFICATIONS

* ID-NBR INI. DATE DESCRIPTION *

* RA1490 RWW 13-MAR-91 INITIAL CREATION OF PROGRAM *

* PL1392I RWW 20-JAN-92 CHANGED CODE WHICH PRINTED
* 'RV' PAGES. THE NEW CODE ONLY*
* PRINTS THE WRITTEN FOR THAT *
* INFORCE AND NUMBERS THE PAGES.*

* PL8492I RWW 03-MAR-92 CHANGED CODE TO ONLY SELECT *
* VINT ROWS WHEN THE *
* VINT_GB_EXP_DATE IS NOT EQUAL *
* VINT_GB_EFF_DATE. ALSO, THE USER*
* TRIED TO BLANK OUT A DBA NAME.*
* THE PROGRAM STILL PRINTED THE *
* BSOR_GB_ADNAME. THE PROBLEM *
* WAS THAT THE PROGRAM ONLY PRINTS THE*
* POLY DBA NAME INSTEAD OF THE *
* BSOR_GB_ADNAME WHEN THE POLY *
* DBA NAME IS NOT SPACES. SO, *
* IF THE FIRST POSITION OF THE *
* POLY NAME OR POLY DBA NAME IS *
* A '*', THEN THE NAME SHOULD BE*
* SPACES. *

* PL12992I 03-APR-92 RWW CHANGED THE LOGIC IN THE 3400 *
* PARAGRAPH TO ONLY ADD THE *
* WS-EXIT-GB-WRIT-PREM TO *
* THE WS-TBL-TL-PREM WHEN THAT *
* COVERAGE TYPE IS PRINTED ON *
* THE SCHEDULE. *

* PL12592I 08-APR-92 RWW ADDED AN ADDITIONAL MESSAGE TO PRINTED *
* WHEN A VEHICLE HAS HAD COVERAGE ADDED *
* OR DELETED. *

* PL18982I 10-APR-92 RWW ADDED CODE TO PRINT RADUIS CLASS FOR *
* VEHICLES MANUALLY RATED, CHANGED LOSS *
* PAYEE SCHEDULE TO CREATE A NEW SCHEDULE*
* WHEN THE STATE CHANGES, AND ADDED CODE *

```

57      *          TO PRINT PRINT THE COMP LIMIT FOR SPEC.*
58      *          PERILS.                                *
59      *          *                                       *
60      * PL19692I  15-APR-92 RWW  ADDED CODE TO ALINE THE PRINTING OF  *
61      *          COVERAGE LIMITS, NOT APPLICABLE TEXT, *
62      *          AND THE DEDUCTIBLE AMOUNT ON THE      *
63      *          REVISION. ALSO, ADDED CODE TO PRINT  *
64      *          THE FULL 20 POSITIONS OF THE CITY ON  *
65      *          THE REVISION.                        *
66      *          *                                       *
67      * PL18892I  30-APR-92 RWW  FIXED CODE TO PRINT THE LIMITS ON A 'RV'*
68      *          WHEN THE ANNUAL AND WRITTEN PREMIUM IS *
69      *          ZERO. ALSO, CHANGED CODE TO NOT PRINT *
70      *          ANYTHING FOR A 'NP' ENDORSEMENT.      *
71      *          *                                       *
72      * PL23592I  04-MAY-92 RWW  INCREASED THE SIZE OF VHCD_AU_CLASS_CODE*
73      *          AND VCLD_AU_CLASS_CODE FROM 5 TO 6.  *
74      *          *                                       *
75      * PL23792I  05-MAY-92 RWW  CHANGED CODE TO MOVE THE PRINTING OF  *
76      *          THE PRT-STATED-AMT VARIABLE 1 POSITION *
77      *          TO THE LEFT.                          *
78      *          *                                       *
79      * PL15492I  07-MAY-92 RWW  CHANGED CODE TO PRINT THE ORTD CODE AND*
80      *          THE ACCOUNT OF THE VESTED INTEREST ON *
81      *          THE SCHEDULE OF LOSS PAYEES.          *
82      *          *                                       *
83      * PL26792I  22-MAY-92 RWW  CHANGED CODE THAT PRINTED THE ORTD CODE*
84      *          ON THE SCEDULE OF LOSS PAYEE TO USE THE*
85      *          ORTD CODE FROM VINT INSTEAD OF BSOR.  *
86      *          *                                       *
87      * PL29292I  05-JUN-92 RWW  CHANGED CODE TO ADD THE PREMIUM IN THE *
88      *          3400 AND 3450 PARAGRAPHS INSTEAD OF  *
89      *          MOVING THE PREMIUM.                  *
90      *          *                                       *
91      * PL33792I  02-JUL-92 RWW  THE OLD CODE WAS MOVING THE WRONG      *
92      *          VARIABLE TO THE TABLE SUBSCRIPT      *
93      *          VARIABLE. THE NEW CODE WILL MOVE      *
94      *          WS-ENTRY-COUNT TO WS-SV-ENTRY-COUNT  *
95      *          INSTEAD OF MOVING WS-VHCD-AU-ID-NO TO *
96      *          WS-LP-CAR-NO. THE WS-LP-CAR-NO WAS LATER*
97      *          FOR A TABLE SUBSCRIPT.              *
98      *          *                                       *
99      * PL35892I  17-JUL-92 RWW  THE PROGRAM WAS PRINTING ALL FORM      *
100     *          NUMBERS ON PAGE 3 OF THE DECLARATION *
101     *          PAGE. HOWEVER, THE PAGE COULD ONLY   *
102     *          HANDLE 24 FORM NUMBERS. SO, ADDED CODE *
103     *          TO PRINT ONLY 24 FORM NUMBERS PER PAGE *
104     *          3. IF MORE THAN 24 FORM NUMBERS EXIST,*
105     *          A NEW PAGE 3 WILL BE PRINTED FOR THE  *
106     *          REMAINING FORM NUMBERS.              *
107     *          *                                       *
108     * PL35192I  11-NOV-92 RWW  REMOVED THE FORM EDITION DATE FROM THE *
109     *          AUTO FORMS. THEREFORE, ADDED CODE TO  *
110     *          THE PROGRAM TO PRINT THE FORM EDITION *
111     *          FROM THE PLFD TABLE.                *
112     *          *                                       *

```

```

113 * PL38392I 26-JAN-93 RWW CHANGE CODE TO COUNT THE LINES PRINTED *
114 *          *          *          *          *          *          *
115 *          *          *          *          *          *          *
116 *          *          *          *          *          *          *
117 *          *          *          *          *          *          *
118 * PL4293I 27-JAN-93 RWW CHANGED CODE TO ONLY PRINT THE OLD *
119 *          *          *          *          *          *          *
120 *          *          *          *          *          *          *
121 *          *          *          *          *          *          *
122 *          *          *          *          *          *          *
123 * PL5193I 02-FEB-93 RWW CHANGED CODE TO PRINT A MESSAGE WHENEVER*
124 *          *          *          *          *          *          *
125 *          *          *          *          *          *          *
126 *          *          *          *          *          *          *
127 *          *          *          *          *          *          *
128 *          *          *          *          *          *          *
129 * PL11893I 22-FEB-93 RWW CHANGED CODE TO PRINT A 'NOTICE OF *
130 *          *          *          *          *          *          *
131 *          *          *          *          *          *          *
132 *          *          *          *          *          *          *
133 * PL10993I 22-FEB-93 RWW ADDED CODE TO PRINT ASSIGNED RISK *
134 *          *          *          *          *          *          *
135 *          *          *          *          *          *          *
136 * PL13793I 01-MAR-93 RWW MOVE CODE WHICH STRINGED THE CITY, *
137 *          *          *          *          *          *          *
138 *          *          *          *          *          *          *
139 *          *          *          *          *          *          *
140 *          *          *          *          *          *          *
141 *          *          *          *          *          *          *
142 *          *          *          *          *          *          *
143 * PL14493I 04-MAR-93 RWW FIXED THE VINT_CURS. THE WHERE CLAUSE *
144 *          *          *          *          *          *          *
145 *          *          *          *          *          *          *
146 *          *          *          *          *          *          *
147 *          *          *          *          *          *          *
148 * PL53192I 05-MAR-93 RWW ADDED CODE TO THE PROGRAM TO ALLOW FOR *
149 *          *          *          *          *          *          *
150 *          *          *          *          *          *          *
151 *          *          *          *          *          *          *
152 *          *          *          *          *          *          *
153 *          *          *          *          *          *          *
154 *          *          *          *          *          *          *
155 *          *          *          *          *          *          *
156 * PL54292I 05-MAR-93 RWW ADDED CODE TO PRINT REWRITE ABOVE THE *
157 *          *          *          *          *          *          *
158 *          *          *          *          *          *          *
159 *          *          *          *          *          *          *
160 * PL57192I 02-APR-93 RWW ADDED CODE TO PRINT INFORMATION ON THE *
161 *          *          *          *          *          *          *
162 *          *          *          *          *          *          *
163 * PL24393I 26-APR-93 RWW CHANGED CODE TO ALSO PRINT LOSS PAYEES *
164 *          *          *          *          *          *          *
165 *          *          *          *          *          *          *
166 *          *          *          *          *          *          *
167 *          *          *          *          *          *          *
168 *          *          *          *          *          *          *

```

```

169 * PL21093I 04-MAY-93 RWW CHANGED CODE TO PRINT 3 COVERED AUTO *
170 *          SYMBOLS AND THE REST ON THE NEXT LINE. *
171 *          *
172 * PL52992I 08-JUN-93 RWW CHANGED CODE TO USE VHDS_AU_NO INSTEAD *
173 *          OF VHDS_AU_SERIAL_NO AS UNID. *
174 *          *
175 * PL33893I 25-JUN-93 RWW CHANGED CODE TO ORDER VEHICLES BY *
176 *          STATE AND THEN CAR NUMBER INSTEAD OF *
177 *          STATE, LOCT NUMBER AND THEN CAR NUMBER.*
178 *          *
179 * PL35493I 12-JUL-93 RWW ADDED CODE TO PRINT COVERAGE CODES *
180 *          'AAPIP' AND 'AUI'. *
181 *          *
182 * PL36893I 22-JUL-93 RWW FIXED CODE WHICH PRINTED THE 'RV' *
183 *          DEC PAGE. THE CODE WAS NOT PRINTING *
184 *          VEHICLES CORRECTLY WHEN THEY CHANGED *
185 *          LOCATIONS. *
186 *          *
187 * PLI42093 26-AUG-93 RWW CHANGED CODE TO ABORT THE PROGRAM *
188 *          WHENEVER AN ERROR OCCURRS. *
189 *          *
190 * PLI46393 24-SEP-93 RWW ADDED CODE TO DECODE AMED TO MED. *
191 *          *
192 * PL32893I 19-OCT-93 RWW ADDED CODE TO PRINT NEW FACTORS FOR IL.*
193 *          *
194 * PLI53193 03-NOV-93 RWW CHANGED THE WORDING WHICH PRINT PIP *
195 *          COVERAGE ON THE RV PAGE AND LP PAGE. *
196 *          *
197 * PLI55493 15-NOV-93 RWW ADDED CODE TO PRINT COVERAGE "SP". *
198 *          *
199 * PLI56893 18-NOV-93 RWW ADDED CODE TO INCLUDE PREMIUM FOR 064 *
200 *          AND 014 IN THE COMP PREMIUM. *
201 *          *
202 * PLI57793 23-NOV-93 RWW MOVE WS-TBL-LP-EXP-DATE FROM THE *
203 *          VEHICLE TABLE TO LOSS PAYEE INTERNAL *
204 *          TABLE BECAUSE A VEHICLE MAY HAVE MORE *
205 *          THAN ONE LOSS PAYEE. *
206 *          *
207 * PLI55993 07-DEC-93 RWW ADDED CODE TO USE VCLD_AU_EFF_DATE. *
208 *          *
209 * PLI17194 30-MAR-94 RWW FIXED TO MOVE WS-HOLD-PPRT-INFC-GB-NO *
210 *          TO WS-PPRT-INFC-GB-NO AFTER THE 2625 *
211 *          PARAGRAPH IS PERFORMED. IF NO ROWS *
212 *          WERE FETCHED FROM THE AUTO_CURS, THE *
213 *          PROGRAM WAS NOT RESETING THE PPRT INFC *
214 *          WHICH WAS CAUSING THE ROW NOT TO BE *
215 *          DELETED FROM THE PPRT TABLE. *
216 *          *
217 * PLI25894 06-MAY-94 RWW FIXED CODE TO INITIALIZE *
218 *          WS-MCTB-GB-CODE & WS-MCTB-GB-EDIT WHEN *
219 *          FCRAD TYPE IS SELECTED. *
220 *          *
221 * PL14292I 10-JUL-94 RWW CHANGED CODE TO USE AUCV-AU-TYPE *
222 *          INSTEAD OF EXIT-WC-CLASS-CODE FOR *
223 *          COVERAGE TYPE. ALSO, ADDED CODE TO *
224 *          PRINT SUB COVERAGES AND CHANGED THE *

```

225	*	METHOD COVERAGE LIMITS ARE SELECTED.	*
226	*		
227	*	PLI46994 30-SEP-94 RWW FIXED CODE WHICH SOMETIMES WAS NOT	*
228	*	PRINTING THE COVERAGE LIMIT ON A 'RV'.	*
229	*		
230	*	PLI50294 13-OCT-94 RWW ADDED CODE TO ALLOW FOR PRINTING OF	*
231	*	GLASS COVERAGE.	*
232	*		
233	*	PLI13895 03-MAR-95 RWW ADDED CODE TO PRINT UMN,UMS,UIN AND UIS*	
234	*	COVERAGE.	*
235	*		
236	*	PLI48594 13-MAR-95 RWW FIXED CODE WHICH WAS NOT PRINTING THE	*
237	*	LIMITS FOR COVERAGE PLANS 2 AND 3.	*
238	*		
239	*	PLI21595 25-APR-95 RWW ADDED CODE TO PRINT NFUM AND ILB.	*
240	*		
241	*	PLI27695 17-JUL-95 RWW ADDED CODE TO HANDLE COMPUTER RATED	*
242	*	ASSESSMENTS.	*
243	*		
244	*	PLI40795 04-AUG-95 RWW FIXED CODE WHICH WAS NOT PRINTING THE	*
245	*	ASSESSMENT TOTAL.	*
246	*		
247	*	PL10792I 18-SEP-95 RWW ADDED CODE TO PRINT "SEE END.." FOR	*
248	*	SPLIT LIMIT POLICIES.	*
249	*		
250	*	PL10792I 25-SEP-95 TCT ADDED CODE TO STRING FORMATTED SPLIT	*
251	*	LIMIT LIABILITY AMOUNTS TOGETHER.	*
252	*		
253	*	PLI50895 06-OCT-95 TCT ADDED LPAD "0" TO SELECT OF AUCV-AU-	*
254	*	LIMIT SO IT WOULD BE 'NUMERIC'.	*
255	*		
256	*	PLI41695 04-DEC-95 RWW ADDED CODE TO PRINT 'NY' PIP LIMITS.	*
257	*		
258	*	PLI41695 05-DEC-95 RWW REFORMATED PRINT LINE FOR NY PIP.	*
259	*		
260	*	PLI10796 04-MAR-96 KRD MODIFIED WS-PLFD-GB-MACRO AND	*
261	*	WS-PRINT-MACRO FROM PIC X(07)	*
262	*	TO PIC X(09).	*
263	*		
264	*	PLI41695 16-APR-96 TCT ADDED CMPWG AND COLWG TO ORDER BY.	*
265	*		
266	*	PLI22696 30-APR-96 TCT ADDED CODE TO SELECT TERR. CODE IF	*
267	*	NOT 'FCPOP' IN 2625 PARAGRAPH.	*
268	*		
269	*	PLI44896 28-OCT-96 MJRIO INCREASED SIZE OF POLICY-INFO VAR	*
270	*	TO ACCOMODATE FOR EXTRA LONG BSTP TYPE	*
271	*	DESCRIPTIONS.	*
272	*		
273	*	PLI30497 20-OCT-97 TCT ADDED UIOT TO DECODES FOR UI COVERAGES.*	
274	*		
275	*	PLI30497 24-OCT-97 TCT ADDED UIMIN, UMOT, UICCIN AND UICCOT	*
276	*	TO DECODES FOR COVERAGES.	*
277	*		
278	*	PLI30897 28-OCT-97 KRD CHANGED PIC SIZE FOR PRT-COVERAGE-LIMIT*	
279	*	WS-PRT-RV-COV-LT AND WS-NOTICE-LIMIT-AMT	
280	*	TO ALLOW PRINTING OF \$1 MILLION	*

281	*	LIABILITY. ALSO ADDED P_ALTER_NLS_YYYY*	
282	*	TO 1000 PARAGRAPH.	*
283	*		*
284	*	PLI30497 17-DEC-97 TCT ADDED UMINPD, UMOTPD, UMUIIN, UMUIOT	*
285	*	TO DECODES FOR COVERAGES.	*
286	*		*
287	*	PLI198 05-JAN-98 DLR ADDED UIIN,UIPDOT,UIPDIN	*
288	*	TO DECODES FOR COVERAGES.	*
289	*		*
290	*	PLI7898 25-FEB-98 DRL ADDED UMSIN, UMSOT, UMNIN, UMNOT,	*
291	*	UISIN, UISOT, UININ, UINOT TO	*
292	*	DECODES FOR COVERAGES.	*
293	*		*
294	*	PLI12298 30-APR-98 DRL ADDED CODE TO ALLOW FOR NEGATIVE	*
295	*	SURCHARGES. ADDED PAGE 2A.	*
296	*		*
297	*	DBA 13-MAY-98 MJR CORRECTED PROBLEM WITH AUCOV2!	*
298	*		*
299	*		*
300	*	PLI13398 13-MAY-98 DLR EXPANDED THE FOLLOWING FIELDS TO 36	*
301	*	WS-CLIENT-CITY-ST-ZIP	*
302	*	WS-LP-CITY-ST-ZIP	*
303	*	WS-SCH-LP-CITY-ST-ZIP	*
304	*	TRUNCATION WAS HAPPENING	*
305	*		*
306	*	PLI12298 27-MAY-98 DRM CORRECTED PROBLEM WITH LINES ADVANCING	*
307	*	ON NON-SIMPLIFIED PAGE 2	*
308	*		*
309	*	PLI20898 14-AUG-98 LAP YEAR 2000 CHANGES	*
310	*		*
311	*	PLI37398 07-DEC-98 LAP ADJUSTED PRINTOUT OF AUTO-NO & YEAR	*
312	*		*
313	*	PLI38398 28-DEC-98 DRM ADDED 'INTER' CODE TO PRINT ON PAGE	*
314	*	2A.	*
315	*		*
316	*	PLI1499 18-JAN-99 FJL RAISED THE ERR TABLE OCCURANCE TO 100	*
317	*	FROM 20 AND ADDED SOME ERROR TRAPPING	*
318	*	TO HELP PREVENT TIME LOST FINDING THIS	*
319	*	TYPE OF ERROR.	*
320	*		*
321	*	PLI6899 09-MAR-99 DRM ADDED NOUM, NOUI, HUM, AND HUI TO	*
322	*	DECODES FOR UM AND UI COVERAGES.	*
323	*		*
324	*	PLI5299 10-MAY-99 LAP AMENDED CODE TO SELECT STATE SPECIFIC	*
325	*	OR XX ENDORSMENT FORM - AMA-104	*
326	*		*
327	*	PLI17399 25-MAY-99 FJL ADDED TWO NEW PIP CODES TO THE DECODES	*
328	*	(PIPST, PIPI) TO INCLUDE THEM IN WITH	*
329	*	THE REGULAR PIP COVERAGE.	*
330	*		*
331	*	PLI18799 08-JUN-99 FJL / JRK ADDED A NEW AUCV_AU_TYPE TO THE	*
332	*	TWO DECODES (UMBI)	*
333	*		*
334	*	PLI28999 30-AUG-99 LAP ADDED NEW AUCV_AU_TYPES TO TWO DECODES	*
335	*	PIPF/PIPO (PIP), ADPIP/OP1-ADPIP/02	*
336	*	(ADPIP)	*


```

337 *
338 * PLI38899 23-NOV-99 FJL ADDED A AUCV_AU_TYPE TO THE DECODE *
339 * 'UMPD' TO 'UM' *
340 *
341 * PLI39399 08-DEC-99 LAP ADDED CODE TO PRINT SPLIT LIMITS FOR *
342 * "HI" *
343 *
344 * PLI41999 21-DEC-1999 FJL ADDED GARKEE TO A IF STATEMENT SO *
345 * COVERAGE WILL PRINT ON PAGE 2A *
346 * (GARAGEKEEPERS COVERAGE0 *
347 *
348 * PLI7000 18-FEB-2000 DRM MODIFIED PARA 7680 TO JUST FORMAT *
349 * LIMIT WHEN THERE ARE 3 ARGUMENTS IN A *
350 * SPLIT LIMIT AND ADDED PARA 7685 TO *
351 * HANDLE CASES WHEN THERE ARE ONLY 2. *
352 * WHEN THE LIABILITY LIMIT IS SPLIT, *
353 * 'SEE POLICY' PRINTS ON AMA-111 NOW. *
354 *
355 * PLI7000 02-MAR-2000 DRM FIXED PROBLEM CREATED BY LAST CHANGE.*
356 * PROGRAM USED UM VARIABLE TO DECIDE *
357 * WHETHER OR NOT RO PRINT UI INFORMATION *
358 * ON A REVISION. *
359 *
360 * PLI9300 15-MAR-2000 DRM BACKED OUT CHANGES FROM PLI38899 *
361 * PER DEANN. *
362 *
363 * PLI10500 28-MAR-2000 DRM MADE CHANGES TO 7225 PARAGRAPH TO *
364 * KEEP LAST 'O' FROM DROPPING FROM *
365 * SPLIT UM & UI LIMITS. *
366 *
367 * PLI13700 22-MAR-2000 DRM STORED SPL LIMIT (BI) IN VARIABLE SO *
368 * THAT IT WOULD PRINT CORRECTLY ON THE *
369 * AMA-104. *
370 *
371 * PLI13900 23-MAR-2000 DRM STORED SPLPD LIMIT (PD) IN VARIABLE *
372 * AND CHANGED 7680 PARA SO THAT THE *
373 * SPLIT LIMITS WILL PRINT CORRECTLY IF *
374 * THE SPL PER PERSON LIMIT IS DIFFERENT*
375 * FROM THE SPLPD LIMIT (WHICH, PER RAY *
376 * DANIELS, IS POSSIBLE). *
377 *
378 * PLI17800 18-JUL-2000 DRM ADDED PIPOB TO DECODE STATEMENTS. *
379 *
380 * PLI18500 26-JUL-2000 DRM CORRECTED PIP TOTAL ON REVISION DEC *
381 * PAGE. PIP ANNUAL NOW INCLUDES OBEL. *
382 *
383 * PLI21500 24-AUG-2000 DRM ADDED UMBIPD AND UIBIPD TO DECODES. *
384 *
385 * PLI16800 13-SEP-2000 LAP ADDED CODED TO PRINT MACROS FOR AUDEC*
386 * AUCOV, AUEND & AULOSS BASED ON THE *
387 * INFC_GB_PRIMARY_STATE INSTEAD OF *
388 * WS_CLIENT_STATE. *
389 * DOES NOT EFFECTIVE AUSCHED *
390 *
391 * PLI26900 03-OCT-2000 LAP INCREASED SIZE OF VARIABLE *
392 * WS-PRT-RV-AUTO-END-NO TO 4 *

```

```

393 *
394 * PLI5901 26-FEB-2001 DRM INCREASED OCCURS FROM 500 TO 1000 *
395 * FOR TWO TABLES: *
396 * WS-SCHEDULE-AUTO-TABLE AND *
397 * WS-SCHEDULE-WRIT-PREM-TABLE. *
398 *
399 * PLI7201 09-MAR-2001 DRM MODIFIED PRT-OLD-AUTO-LINE SO THAT *
400 * THE AUTO ID NUMBERS PRINT INSIDE THE *
401 * BOX ON THE JETFORM. *
402 *
403 * PLI7201 12-MAR-2001 DRM MODIFIED PRT-OLD-AUTO-LINE (AGAIN) *
404 * SO THAT THE CITY AND TERRITORY PRINT *
405 * INSIDE THE BOX ON THE JETFORM. *
406 *
407 * PLI8901 22-MAR-2001 DRM ADDED SPLIT LIMITS INDICATORS FOR *
408 * EACH COVERAGE THAT CAN BE SPLIT. *
409 * ALSO REMOVED REDUNDANT DECODES OF *
410 * 'ADPIP'. *
411 *
412 * PLI24201 26-JUN-2001 JJB ADDED WS-PRINT-PAGE-2A, *
413 * WS-CREDIT-COUNT AND WS-ASSM-TABLE *
414 * TO 5000-REINIT-VAR TO ALLOW THEM TO *
415 * BE CLEARED EACH TIME THROUGH. *
416 *
417 * PLI10801 28-JUN-2001 DRM CHANGED VINT_CURS SO THAT VESTED *
418 * INTERESTS WITH ORTD OF AIO DO NOT *
419 * PRINT. THESE JUST MAKE NY ACCEPT *
420 * OUR EDI DATA. *
421 *
422 * PLI39901 08-NOV-2001 JJB CHANGED TO #1 HORTICULTURAL LANE *
423 * IN 7650-PRINT-NOTICE. *
424 *
425 * PLI43001 28-NOV-2001 AAP REMOVED DELETE FROM PPRT. PPRT ROW *
426 * WILL BE DELETED IN ANOTHER PROGRAM. *
427 * PLI20602 06-MAY-2002 BTG ADDED PIP COVERAGES OBWRK, OBAE, *
428 * OBFE, OBDB *
429 *
430 * PLI30000 17-JAN-2003 ANT GUI CONVERSION *
431 *
432 * PLI30000 31-JAN-2003 ANT FIXED PROBLEM WITH TOO MANY ROWS *
433 *
434 * PLI30863.1699 08-APR-2003 LAW ALTERED ATTACHED_CURS TO ONLY SELECT *
435 * POLICY FORM THAT ARE MADATORY "M" OR *
436 * OPTIONAL "O" -- NOT REQUIRED "R" OR *
437 * OPTIONAL "T" *
438 *
439 * PLI31626.3418 08-JAN-2004 AAP ADDED 'VA' SPECIFIC CODE TO HANDLE *
440 * PRINTING VA DECPAGES WITH THE NEW *
441 * COVERAGES MEDICAL EXPENSE AND INCOME *
442 * LOSS BENEFITS. CHANGES INCLUDE *
443 * ADDING 7255-PRINT-VA-SIMPLIFIED AND *
444 * 7260-VA-PRT-ADD-PIP. *
445 *
446 * PLI31528.3599 11-MAR-2004 TCT REMOVED "AND EXIT-AUBP-AU-NO=0" LINE *
447 * FROM 3450-PROC-AUTO-NET-PREM IF STMT.*
448 * THIS WAS CAUSING THE RADIUS FACTOR TO*

```



```

449 * NOT PRINT FOR SOME VEHICLES TIED TO *
450 * EXITS THAT ARE TIED TO 0 AUBP ROWS. *
451 *
452 * PLI31772.3844 07-APR-2004 JPC WIDENED PRT-END-x TO ACCOMODATE NEW, *
453 * LONGER FORM NAMES FOR ENDORSEMENTS *
454 * ALSO ADDED "ABEND:YES/N0" LOGIC. *
455 *
456 * PLI31792.3891 21-APR-2004 AAP CHANGED MED-EXP VARIABLE TO MED-PAY *
457 * VARIABLE IN 7255-PRINT-VA-SIMPLIFIED *
458 * ALSO REMOVED SOME UNNECESSARY CODE *
459 * AND CORRECTED A SPACING PROBLEM *
460 *
461 * PLI31818.3963 14-JUN-2004 AAP REDEFINED SEVERAL PRINT LINES AND *
462 * ADDED SOME CODE CHECKING FOR EXISTANCE*
463 * OF COVERAGES SO WE CAN PRINT ZEROS *
464 * WHEN A COVERAGE EXISTS BUT THERE IS *
465 * NO CHARGE FOR THE COVERAGE. *
466 *
467 * PLI31903.4357 06-OCT-2004 AAP ADD WORDING FOR IL HIRED AUTO AND *
468 * NON-OWNERSHIP COVERAGES TO ONLY PRINT*
469 * WHEN POLICY HAS AN IL LOCATION. ALSO *
470 * ADDED CODE TO SELECT TERRITORY CODE *
471 * FROM NEW TERRITORY_HISTORY TABLE. *
472 * PLI31903.4385 18-OCT-2004 AAP CORRECT DATE FORMATTING ERROR IN *
473 * TERRITORY CODE CURSOR *
474 * PLI31528.4361 20-OCT-2004 TCT CORRECT RADIUS PRINTING FOR NEW IL *
475 * SCHEDULE *
476 * PLI31528.4381 22-OCT-2004 TCT "-exist" VARIABLES NEEDED TO BE *
477 * REINITIALIZED. *
478 *
479 * PLI31612.4325 27-OCT-2004 JPC ADD HOME OFFICE P.O. BOX ADDRESS *
480 *
481 * PLI31612.4416 01-NOV-2004 JPC CORRECT SPACING WHERE SOME WORDS WERE
482 * RUNNING TOGETHER
483 *
484 * PLI31528.4442 10-NOV-2004 TCT CORRECTED NVL ON TRH_GB_EXP_DATE *
485 *
486 * PLI31981.4745 10-JAN-2005 HLC CORRECTED LITERAL @ PARAGRAPH 7260 *
487 *
488 * PLI31993.5298 17-AUG-2005 AAP CHANGED CODE TO PRINT MACROS FOR *
489 * AULOSS BASED ON THE GARAGE STATE *
490 * (WS-SCH-GARAGE-STATE) RATHER THAN *
491 * INFC_GB_PRIMARY_STATE SO IT WILL *
492 * PICK UP THE CORRECT EDITION DATES. *
493 *
494 * PLI32086.5436 30-AUG-2005 AAP ADDED FENAM AND FEDES TO PRINT ON *
495 * PAGE 2A. *
496 *
497 * PLI31990.5547 13-OCT-2005 JPC DECODED 'DUI' TO 'UNFAVORABLE' *
498 *
499 * PLI31993.5597 17-NOV-2005 AAP ADDED CODE TO PUSH TO PAGE 2A FOR *
500 * TXFEE. ALSO REDEFINED PRINT-LINE *
501 * TO ACCOMODATE LENGTH OF DESCRIPTION *
502 *
503 * PLI31993.5620 05-DEC-2005 AAP ADDED CHECK FOR 'SPEC' AUCV TYPE TO *
504 * PRINT UNDER SPECIAL PERILS ON THE *

```

SCHEDULE FOR TX.

*
 *
 * PLI31993.5639 19-DEC-2005 AAP ADDED MCTB CHECK TO DETERMINE WHEN *
 * AN ASSM SHOULD PUSH TO PAGE 2A, *
 * REPLACES HARD CODING THE ASSM CODES.*
 * REMOVED TXFEE, EXPMOD, SIZMOD, *
 * INTER, AND GARKEE THAT WERE *
 * PREVIOUSLY HARD CODED. ADDED STATE *
 * TO ASSM DESC FOR PAGE2A AND A CHECK *
 * TO NOT PRINT STATE FOR POLICY LEVEL *
 * CHARGES. ALSO ADDED SPECIAL WORDING*
 * FOR PIP COVERAGE IN HAWAII. *
 *
 * PLI31993.5761 24-JAN-2006 AAP ADDED A DECODE TO THE PAGE 2A SELECT*
 * SO THAT POLICY LEVEL SURCHARGES WILL*
 * NOT PICK UP THE STATE - THIS WAS *
 * PREVENTING THE AMTS FOR LIKE POLICY *
 * LEVEL COVGS FROM BEING COMBINED AND *
 * PRINTED ON ONE LINE *
 *
 * PLI31974.5782 07-FEB-2006 HLC ADDED 4 AUCV_AU_TYPE TO TWO DECODEs *
 * UUBN, UUBPN, UUBS, UUBPS *
 *
 * PLI32298.6453 05-OCT-2006 JPC ADD NEW CONNECTICUT COVERAGE(S) *
 *
 * PLI32302.6539 09-OCT-2006 JPC ADD HANDLING FOR ISO RATED STATES *
 *
 * PLI31981.6566 13-OCT-2006 HLC Add work field for string delimiter *
 * Failed on city of 18 or 19 long *
 *
 * PLI32502.6768 15-DEC-2006 JPC Add RLOAN to SLL totals and print on*
 * page 2. *
 * PLI32502.6922 18-DEC-2006 JPC WIDENED PRINT LINE
 * PLI32315.6857 27-DEC-2006 TCT FIXED TRHS SELECT NOT HANDLING *
 * LOCTS PROPERLY. *
 * PLI32513.7233 12-FEB-2007 AAP CHANGED LOGIC FOR TERR. CODE TO *
 * ALWAYS PRINT UNLESS ZONE RATED, THEN*
 * 'ZONE' WILL PRINT. *
 * PLI32513.7524 13-FEB-2007 AAP FIX PROBLEM WITH MANUALLY RATED AUTOS*
 * PRINTING 'ZONE' INSTEAD OF TERR. CODE*
 * PLI32513.8425 01-JUN-2007 AAP Added GCOM to AUTO_PREM_CURS to pick*
 * up Glass coverage premium. *
 * PLI32513.6665 02-FEB-2007 AAP IN ORDER TO ACCOMMODATE MULTIPLE *
 * CONFLICTING REQUIREMENTS FROM *
 * DIFFERENT STATES, CHANGED PAGES 2 *
 * AND 2A COMPLETELY VARIABLE. THIS *
 * REQUIRED EXTENSIVE CHANGES TO SECTS *
 * 2390,2396,2600,3400,3450,7100,7200, *
 * 7225,7230,7235,7250,7275 AND 7450 *
 * AND ADDITION OF 2250,2260 AND 2627. *
 * ALSO ADD CODE TO PRINT A PAGE 3A IF *
 * THERE ARE TOO MANY FORMS TO PRINT *
 * IN ITEM SIX OF AMA-101S (PAGE 3) *
 * Joint move PLI32503.7631 HLC Correct wrong limits in Loss-Payee
 * printed from table -but also used non-table
 * residual WS-AUCV-AU-LIMIT-SPLIT-BI & '-HOLD

```

561      * so added elements to the table *
562      * PLI32513.8614 19-JUN-2007 CORRECTED NY PIP AND UNINSURED *
563      * MOTORIST LIMITS AND PREMIUM NOT *
564      * PRINTING ON REVISION POLICY *
565      * PLI32513.8720 02-JUL-2007 AAP Correct OBEL not being included with*
566      * NY PIP premium on revision. *
567      * PLI32513.8755 05-JUL-2007 AAP Reset WS-PRT-2A switch to correct *
568      * line count problem when multiple *
569      * policies are requested together *
570      * PLI32513.7630 20-JUL-2007 AAP Added BROD and LIMO to AUCV_AU_TYPE *
571      * decode. *
572      * PLI32513.9016 14-AUG-2007 AAP Added NCRB line for NC UM coverage. *
573      * *
574      * PLI32513.9365 05-SEP-2007 AAP Cleared out state exist switches *
575      * before fetching new pppt row *
576      * PLI32513.9379 07-SEP-2007 AAP Found a switch that was missed in *
577      * previous change. Also set them to *
578      * 'N' instead of spaces. And added *
579      * 'TUI' to the transporter select. *
580      * PLI32513.9599 08-OCT-2007 AAP Made corrections to print deductible*
581      * and premium for MI BROAD, LIMO, & *
582      * LIMF coverages on the AMA-102, AMA-104
583      * and AMA-111. *
584      * *
585      * PLI32513.9684 12-OCT-2007 AAP Changes to print '$0' for the LIMF *
586      * deductible on the AMA-102 and AMA-104*
587      * *
588      * PLI32513.9809 01-NOV-2007 AAP Added ws-state-count to 5000-reinit-var*
589      * *
590      ***** JOINT MOVE*****
591      * PLI32503.9894 13-NOV-2007 HLC Add TOW coverage for WI *
592      * PLI32513.9894 05-DEC-2007 AAP Added wording for WI UM similar *
593      * to IL in 7300-PRINT-HIRED-NON-OWNED *
594      *****
595      * PLI32513.10063 11-DEC-2007 AAP Cleared WS-ASSM-HOLD & WS-REDUCT to *
596      * correct assessments not printing. *
597      * Combined SIZMOD and EXPRED and set *
598      * to print with new description *
599      * "Expense Modification". *
600      * PLI32513.10540 31-JAN-2008 AAP Moved NY limit code from under BASIC*
601      * to directly under PIP because we are*
602      * no longer using the BASIC sub type *
603      * for NY PIP *
604      * PLI32641.10945 14-MAR-2008 GTM EXIT_GB_NO CHANGES
605      * PLI32641.11431 22-APR-2008 GTM clearing WS-SUM-WRIT-PREM-SUM to
606      * prevent premium from carrying over to another policy
607      * *
608      * PLI32513.7997 14-MAY-2008 AAP Changed header to "RETURN PREMIUMS" *
609      * when a vehicle is deleted and the *
610      * premium amount is zero. *
611      * *
612      * PLI32513.11941 30-JUN-2008 AAP Added GPIIP to decode. *
613      * *
614      * PLI32513.12053 16-JUL-2008 AAP Added PIP deductible for KY *
615      * *
616      * PLI32503.12092 23-JUL-2008 HLC Move 3 lines on AMA-111 to the right*

```

617	*	of the existing lines at the bottom *
618	*	
619	*	PLI32513.12236 12-AUG-2008 AAP Added SUBRO to OTHER_COV_CURS so it *
620	*	will print on decpage *
621	*	
622	*	PLI32510.12392 01-OCT-2008 TCT Added Trailer Interchange logic to *
623	*	OTHER_COV_CURS so it will print *
624	*	
625	*	PLI32510.12606 02-OCT-2008 TCT Removed literal coding from union *
626	*	
627	*	PLI32513.13122 11-DEC-2008 AAP Added city/county name to print for *
628	*	KY municipal tax. *
629	*	Also added new GA UM codes to decode*
630	*	
631	*	PLI32503.14603 01-JUN-2009 HLC Change edit-mask used to print '047'*
632	*	(rental reimbursement) -support 0.00*
633	*	
634	*	PLI32503.15049 29-JUN-2009 HLC Add support for APDTH in NY *
635	*	
636	*	PLI32503.8391 21-AUG-2009 HLC Add support for new AMA-102FL *
637	*	
638	*	PLI32502.16393 09-NOV-2009 JPC Allow zero-dollar printing for Auto *
639	*	Medical Payments *
640	*	PLI32647.16413 13-NOV-2009 MJV Add support for TPIP and TSTUM *
641	*	
642	*	PLI32503.17074 22-JAN-2010 HLC Add support for UIPD in WA *
643	*	
644	*	PLI32502.17526 07-MAY-2010 JPC add UI/UM verbiage and logic for IN *
645	*	PLI32502.18466 17-MAY-2010 JPC Back out previous addition *
646	*	
647	*	PLI32502.17498 20-MAY-2010 JPC Correct linespace issues from 17526 *
648	*	
649	*	PLI32502.18512 21-MAY-2010 JPC add conditional linespacing *
650	*	for IL/WI/IN *
651	*	
652	*	PLI32502.18527 25-MAY-2010 JPC Add print routine for regular states*
653	*	
654	*	PLI32502.18982 26-JUL-2010 JPC Adjust linespacing for forms list *
655	*	
656	*	PLI32502.20512 11-MAR-2011 JPC Add logic for CKCOLL, GKCOMP *
657	*	
658	*	PLI32768.20741 01-APR-2011 REB display the SYSTEM_RATED_IND column *
659	*	from the POLY table *
660	*	
661	*	PLI32502.21936 28-NOV-2011 JPC Add Hired/Non-Owned Medical Payments*
662	*	text and logic *
663	*	
664	*	PLI32502.21257 01-DEC-2011 JPC Text edits *
665	*	
666	*	PLI32502.22187 12-DEC-2011 JPC Add default UI/UM text *
667	*	
668	*	PLI32502.22259 16-DEC-2011 JPC correct various line-spacing issues *
669	*	
670	*	PLI32502.22269 19-DEC-2011 JPC adjust WI line spacing *
671	*	
672	*	PLI32502.22294 13-JAN-2012 JPC allow for RSLL, RMED, RPIP, RUM *


```

673 *
674 * PLI32502.22576 09-FEB-2012 JPC correct Liability limits, Medical and UM lines
675 * PLI32502.22630 13-FEB-2012 JPC backout previous modification *
676 * PLI32502.23487 13-JUL-2012 JPC backout UIUM/HANO modifications
677 * PLI32502.23748 28-AUG-2012 JPC Skip Local, Inter, Long logic when *
678 * 7960,7961,or 7963*
679 *
680 *****JOINT MOVE*****
681 * PLI32502.23694 19-SEP-2012 JPC page layout changes for company 5 *
682 * PLI32503.23695 19-SEP-2012 HLC Company 5: AMA-104, AMA-111 Edition *
683 *****
684 *
685 * PLI32503.23695 01-APR-2013 HLC Company 5: AMA-104, AMA-103 fixes *
686 *
687 * PLI32503.25178 19-APR-2013 HLC fix multipage AULOSS Company Name *
688 *
689 * PLI32503.25242 30-APR-2013 HLC Improve AMA-104 alignment AMA-111 *
690 * Heading *
691 *
692 * PLI32502.25579 20-JUN-2013 JPC Change criteria for MI text *
693 * PLI32502.25568 03-JUL-2013 JPC Add handling for TXSD coverages *
694 * PLI32502.24495 01-AUG-2013 JPC Make corrections for TXSD
695 * PLI32502.25897 13-AUG-2013 JPC added cutoff date for IN UM/UI logic
696 * PLI32502.26574 18-NOV-2013 JPC restrict addition of DSMED *
697 * PLI32502.26898 10-JAN-2014 JPC Adjust PRT-PPI-WRIT alignment *
698 * PLI32508.27021 25-feb-2014 kgm Added requestor id page *
699 * PLI32502.27438 10-APR-2014 JPC Replace 'Bypass'
700 * PLI32502.27489 16-APR-2014 JPC Added DSPIP 3450-proc-auto-net-prem
701 * PLI32502.27521 18-APR-2014 JPC Correct DSPIP placement
702 * PLI32502.27669 14-MAY-2014 JPC Add HAMED, NOMED to 7302-DETERMINE-COVG
703 * PLI32502.27807 11-JUN-2014 JPC Added GCOL to AUTO_PREM_CURS to pick*
704 * up Glass coverage premium. *
705 * PLI32502.27755 19-JUN-2014 JPC add IF/THEN to 7550 block *
706 * PLI32502.27870 25-JUN-2014 JPC Fix AMA-103 logic to allow multi- *
707 * state Loss Payees *
708 * PLI32502.28069 31-JUL-2014 JPC Correct spacing for WI HA/NO text *
709 * PLI32876.27062 03-NOV-2014 LAW Adding additional plate coverage *
710 * PLI32502.31092 12-MAY-2016 JPC MI, NJ, KY; ADPIP logic
711 * PLI32502.30807 16-MAY-2016 JPC Add PEDPIP
712 * PLI32502.31128 18-MAY-2016 JPC Recompile for orasql11
713 * PLI32502.31081 26-MAY-2016 JPC Add "WHEN 'DESMED'"
714 * PLI32503.31341 01-AUG-2016 HLC Reset index to prevent ABEND (prt OK*
715 * PLI32502.31815 31-OCT-2016 JPC Increase all indices
716 * PLI32876.31855 03-NOV-2016 LAW Adding more registration plate codes
717 * and EXAPIP/EXPIP
718 * PLI32502.31852 07-DEC-2016 JPC Add 047CP/CL *
719 * PLI32503.31759 08-DEC-2016 HLC Support HAUM/NOUM for Vermont *
720 * PLI32502.32061 20-DEC-2016 JPC Refine WS-NYAPIP-EXIST IF/THEN logic
721 * PLI32502.32159 16-JAN-2017 JPC Additional NY logic
722 * PLI32502.31498 17-JAN-2017 JPC AMA-111 changes; program towing limit to print
723 * PLI32502.32176 26-JAN-2017 JPC back out previous
724 * PLI32502.31509 07-FEB-2017 JPC Set Towing charge to $50
725 * PLI32502.32442 12-APR-2017 JPC Add MCSLL to exposure lists
726 * PLI32876.32412 19-JUN-2017 LAW Adding cycy_history
727 * PLI32502.32729 07-AUG-2017 JPC Print WAIVE on California policies
728 * PLI32502.32730 12-OCT-2017 JPC Print GLASS when needed

```

```

729 * PLI32502.33206 20-OCT-2017 JPC Alter GLASS logic for GCOL/GCOM
730 * PLI32502.32757 13-OCT-2017 JPC Print RENTAL when needed
731 *-----*
732 *                ENVIRONMENT DIVISION                *
733 *-----*
734 *
735 ENVIRONMENT DIVISION.
736 CONFIGURATION SECTION.
737 *
738 SOURCE-COMPUTER.  VAX-8350.
739 OBJECT-COMPUTER.  VAX-8350.
740 *
741 SPECIAL-NAMES.
742     ARGUMENT-NUMBER IS AUG-NUM
743     ARGUMENT-VALUE IS AUG-VAL
744     CONSOLE IS SCREEN-INPUT.
745 *
746 INPUT-OUTPUT SECTION.
747 FILE-CONTROL.
748 *
749 *     SELECT REPORT-OUT ASSIGN TO "SYS$LOGIN:AUDEC.PRT".
750 *     SELECT REPORT-OUT ASSIGN TO WS-REPORT-OUT.
751 *
752 *-----*
753 *                DATA DIVISION                *
754 *-----*
755 *
756 DATA DIVISION.
757 FILE SECTION.
758 *
759 FD REPORT-OUT  DATA RECORD IS PRINT-LINE.
760 01 PRINT-LINE.
761     05 PRT-ACCT-LINE.
762         10 FILLER                PIC X(15).
763         10 PRT-POLY-NO            PIC X(15).
764         10 FILLER                PIC X(34).
765         10 PRT-ACCT-NO            PIC ZZZZZZZ.
766         10 FILLER                PIC X(170).
767     05 PRT-NAME-LOCT-LINE REDEFINES PRT-ACCT-LINE.
768         10 FILLER                PIC X(19).
769         10 PRT-NAME-LOCT-INFO     PIC X(60).
770         10 FILLER                PIC X(61).
771     05 PRT-POLICY-LINE REDEFINES PRT-ACCT-LINE.
772         10 FILLER                PIC X(18).
773         10 PRT-POLICY-INFO        PIC X(40).
774         10 FILLER                PIC X(99).
775     05 PRT-PA-HIRPHY-LINE REDEFINES PRT-ACCT-LINE.
776         10 FILLER                PIC X(03).
777         10 PRT-PA-VERB            PIC X(75).
778         10 FILLER                PIC X(79).
779     05 PRT-POLICY-COSIGN REDEFINES PRT-ACCT-LINE.
780         10 FILLER                PIC X(18).
781         10 PRT-POLICY-DATE        PIC X(10).
782         10 FILLER                PIC X(04).
783         10 PRT-FM-CITY            PIC X(53).
784         10 FILLER                PIC X(57).

```



```

785 05 PRT-SUM-LINE-SPLIT REDEFINES PRT-ACCT-LINE.
786     10 FILLER PIC X(01).
787     10 PRT-SUM-DESCRIPTION1 PIC X(51).
788     10 PRT-SUM-DESCRIPTION2 PIC X(60).
789     10 FILLER PIC X(35).
790 05 PRT-SUM-LINE REDEFINES PRT-ACCT-LINE.
791     10 FILLER PIC X(01).
792     10 PRT-SUM-DESCRIPTION PIC X(110).
793     10 FILLER PIC X(35).
794 05 PRT-COVERAGE-LINE REDEFINES PRT-ACCT-LINE.
795     10 FILLER PIC X(01).
796     10 PRT-COV-DESCRIPTION.
797         15 PRT-COVERAGE-DESC PIC X(35).
798         15 PRT-COVERAGE-AUTO PIC X(05).
799     10 PRT-ASSM-COVERAGE REDEFINES PRT-COV-DESCRIPTION.
800         15 PRT-ASSM-COV-DESC PIC X(35).
801     10 FILLER PIC X(01).
802     10 PRT-COVERAGE-AMOUNT.
803         15 PRT-COVERAGE-LIMIT-A.
804             20 PRT-COVERAGE-LIMIT PIC $$$,$$$,$$$$.
805         15 PRT-COVERAGE-LIMIT-B REDEFINES PRT-COVERAGE-LIMIT-A.
806             20 PRT-COVERAGE-LIMIT-CHAR PIC X(08).
807         15 FILLER PIC X(01).
808         15 PRT-COVERAGE-LIMIT-C.
809             20 PRT-DOLLAR-SIGN-2 PIC X(01).
810             20 PRT-COVERAGE-LIMIT-2 PIC Z,ZZZ,ZZZ.
811         15 PRT-COVERAGE-LIMIT-D REDEFINES PRT-COVERAGE-LIMIT-C.
812             20 PRT-COVERAGE-LIMIT-2-CHAR PIC X(09).
813         15 PRT-COVERAGE-TEXT PIC X(18).
814     10 PRT-COVERAGE-END REDEFINES PRT-COVERAGE-AMOUNT.
815         15 PRT-COVERAGE-SEP-STATED PIC X(28).
816     10 PRT-PIP-DEDUCTIBLE REDEFINES PRT-COVERAGE-AMOUNT.
817         15 PRT-PIP-COV-DESC PIC X(01).
818         15 PRT-PIP-DEDUCT PIC ----.
819         15 PRT-PIP-DESC PIC X(23).
820     10 PRT-COVERAGE-COST.
821         15 PRT-COST-DECIMAL PIC -,---,---.---.
822     15 PRT-ZERO-COST-DECIMAL REDEFINES PRT-COST-DECIMAL
823         PIC -,---,---9.99.
824     15 PRT-ZERO-SUB-AUCV-COST REDEFINES
825         PRT-ZERO-COST-DECIMAL PIC Z,ZZZ,ZZ9.99.
826     15 PRT-SUB-AUCV-COST REDEFINES
827         PRT-ZERO-SUB-AUCV-COST PIC Z,ZZZ,ZZZ.ZZ.
828     10 PRT-COVERAGE-COST-NO-DEC REDEFINES PRT-COVERAGE-COST.
829         15 PRT-COST-NO-DEC PIC -,---,---.
830     10 FILLER PIC X(50).
831 05 PRT-HRD-NONOWN-LINE REDEFINES PRT-ACCT-LINE.
832     10 FILLER PIC X(65).
833     10 PRT-HN-COST-DECIMAL PIC -,---,---.---.
834     10 FILLER PIC X(50).
835 05 PRT-LONG-2A-DESC REDEFINES PRT-ACCT-LINE.
836     10 FILLER PIC X(01).
837     10 PRT-2A-COVERAGE-DESC PIC X(102).
838 05 PRT-IL-COVERAGE-LINE REDEFINES PRT-ACCT-LINE.
839     10 FILLER PIC X(04).
840     10 PRT-IL-COVERAGE-DESC PIC X(115).

```

```

841 05 PRT-ASSM-LINE REDEFINES PRT-ACCT-LINE.
842 10 FILLER PIC X(03).
843 10 PRT-2A-ASSM-COV-DESC PIC X(60).
844 10 FILLER PIC X(08).
845 10 PRT-ASSM-COST-DECIMAL PIC -,---,---.---.
846 10 PRT-ZERO-ASSM-COST-DECIMAL REDEFINES
847 PRT-ASSM-COST-DECIMAL PIC -,---,--9.99.
848 10 FILLER PIC X(83).
849 05 PRT-KY-ASSM-LINE REDEFINES PRT-ACCT-LINE.
850 10 FILLER PIC X(03).
851 10 PRT-KY-ASSM-COV-DESC PIC X(16).
852 10 PRT-ASSM-DASH PIC X(03).
853 10 PRT-KY-CITY-CNTY PIC X(22).
854 10 FILLER PIC X(02).
855 10 PRT-KY-ASSM-FEE PIC X(25).
856 10 PRT-KY-ASSM-COST-DECIMAL PIC -,---,---.---.
857 10 PRT-KY-ZERO-ASSM-COST-DECIMAL REDEFINES
858 PRT-KY-ASSM-COST-DECIMAL PIC -,---,--9.99.
859 10 FILLER PIC X(83).
860 05 PRT-AGGREGATE-LINE REDEFINES PRT-ACCT-LINE.
861 10 FILLER PIC X(01).
862 10 PRT-AGGR-DESC PIC X(31).
863 10 FILLER PIC X(10).
864 10 PRT-SUM-TL-PIP-LT PIC $$$,$$$,$$$$.
865 10 FILLER PIC X(29).
866 10 PRT-SUM-TOTAL-PIP PIC Z,ZZZ,ZZZ.ZZ.
867 10 PRT-ZERO-SUM-TOTAL-PIP REDEFINES PRT-SUM-TOTAL-PIP
868 PIC Z,ZZZ,ZZ9.99.
869 05 PRT-COVERAGE-LINE-SM REDEFINES PRT-ACCT-LINE.
870 10 FILLER PIC X(01).
871 10 PRT-COVERAGE-DESC-SM PIC X(35).
872 10 FILLER PIC X(06).
873 10 PRT-COVERAGE-LIMIT-SM PIC $$$,$$$,$$$$.
874 10 FILLER PIC X(18).
875 10 PRT-SUB-AUCV-COST-SM PIC Z,ZZZ,ZZZ.ZZ.
876 10 PRT-ZERO-SUB-AUCV-COST-SM REDEFINES
877 PRT-SUB-AUCV-COST-SM PIC Z,ZZZ,ZZ9.99.
878 10 PRT-COVERAGE-TEXT-SM PIC X(15).
879 10 PRT-COVERAGE-COST-SM.
880 15 PRT-COST-DECIMAL-SM PIC Z,ZZZ,ZZZ.ZZ.
881 10 PRT-ZERO-COVERAGE-COST-SM REDEFINES
882 PRT-COVERAGE-COST-SM.
883 15 PRT-ZERO-COST-DECIMAL-SM PIC Z,ZZZ,ZZ9.99.
884 10 PRT-COVERAGE-COST-NO-DEC-SM
885 REDEFINES PRT-ZERO-COVERAGE-COST-SM.
886 15 PRT-COST-NO-DEC-SM PIC Z,ZZZ,ZZZ.
887 10 FILLER PIC X(60).
888 05 PRT-END-LINE REDEFINES PRT-ACCT-LINE.
889 10 FILLER PIC X(05).
890 10 PRT-END-1 PIC X(25).
891 10 FILLER PIC X(05).
892 10 PRT-END-2 PIC X(25).
893 10 FILLER PIC X(05).
894 10 PRT-END-3 PIC X(25).
895 10 FILLER PIC X(05).
896 10 PRT-END-4 PIC X(25).

```

```

897      10 FILLER PIC X(26).
898 05 PRT-AUTO-LINE REDEFINES PRT-ACCT-LINE.
899      10 FILLER PIC X(04).
900      10 PRT-AUTO-NO PIC ZZZZ.
901      10 FILLER PIC X(02).
902      10 PRT-YEAR PIC X(04).
903      10 FILLER PIC X(01).
904      10 PRT-MAKE-MODEL PIC X(27).
905      10 FILLER PIC X(02).
906      10 PRT-SERIAL-NO PIC X(20).
907      10 FILLER PIC X(01).
908      10 PRT-POP-DEN PIC X(15).
909      10 FILLER PIC X(01).
910      10 PRT-DRIVER-REC PIC X(12).
911      10 FILLER PIC X(02).
912      10 PRT-MILEAGE PIC X(07).
913      10 FILLER PIC X(05).
914      10 PRT-USE PIC X(02).
915      10 FILLER PIC X(03).
916      10 PRT-WEIGHT PIC X(04).
917      10 FILLER PIC X(02).
918      10 PRT-CAR-COST PIC ZZZZZZZ.
919      10 FILLER PIC X(02).
920      10 PRT-LOSS-PAYEE PIC X(03).
921 05 PRT-OLD-AUTO-LINE REDEFINES PRT-ACCT-LINE.
922      10 FILLER PIC X(03).
923      10 PRT-OLD-AUTO-NO PIC ZZZZ.
924      10 FILLER PIC X(02).
925      10 PRT-OLD-YEAR PIC X(04).
926      10 FILLER PIC X(03).
927      10 PRT-OLD-MAKE-MODEL PIC X(32).
928      10 FILLER PIC X(01).
929      10 PRT-OLD-SERIAL-NO PIC X(20).
930      10 FILLER PIC X(01).
931      10 PRT-OLD-TERR PIC X(15).
932      10 FILLER PIC X(01).
933      10 FILLER PIC X(02).
934      10 PRT-OLD-MILEAGE PIC X(07).
935      10 FILLER PIC X(02).
936      10 PRT-OLD-USE PIC X(02).
937      10 FILLER PIC X(06).
938      10 PRT-OLD-WEIGHT PIC X(04).
939      10 FILLER PIC X(02).
940      10 PRT-OLD-CAR-COST PIC ZZZZZZZ.
941      10 FILLER PIC X(08).
942      10 PRT-OLD-LOSS-PAYEE PIC X(03).
943 05 PRT-PREM-LINE REDEFINES PRT-ACCT-LINE.
944      10 FILLER PIC X(03).
945      10 PRT-PREM-NO PIC ZZZZ.
946      10 FILLER PIC X(02).
947      10 PRT-LIAB-WRIT PIC ZZ,ZZZ.
948      10 PRT-ZERO-LIAB-WRIT REDEFINES
949          PRT-LIAB-WRIT PIC ZZ,ZZ9.
950      10 FILLER PIC X(01).
951      10 PRT-BI-WRIT PIC ZZ,ZZZ.
952      10 PRT-ZERO-BI-WRIT REDEFINES PRT-BI-WRIT PIC ZZ,ZZ9.

```

```

953      10 FILLER                      PIC X(02).
954      10 PRT-PD-WRIT                  PIC ZZ,ZZZ.
955      10 PRT-ZERO-PD-WRIT REDEFINES PRT-PD-WRIT PIC ZZ,ZZ9.
956      10 FILLER                      PIC X(02).
957      10 PRT-PIP-WRIT                 PIC ZZ,ZZZ.
958      10 PRT-ZERO-PIP-WRIT REDEFINES
959          PRT-PIP-WRIT                 PIC ZZ,ZZ9.
960      10 PRT-ADDED-PIP-WRIT           PIC ZZ,ZZZ.
961      10 PRT-ZERO-ADDED-PIP-WRIT REDEFINES
962          PRT-ADDED-PIP-WRIT          PIC ZZ,ZZ9.
963 *****-- 10 FILLER                  PIC X(01).
964      10 PRT-PPI-WRIT                  PIC ZZ,ZZZ.
965      10 PRT-ZERO-PPI-WRIT REDEFINES PRT-PPI-WRIT PIC ZZ,ZZ9.
966      10 FILLER                      PIC X(02).
967      10 PRT-MED-PAY-WRIT             PIC ZZ,ZZZ.
968      10 PRT-ZERO-MED-PAY-WRIT REDEFINES
969          PRT-MED-PAY-WRIT           PIC ZZ,ZZ9.
970      10 PRT-UM-WRIT                  PIC ZZ,ZZZ.
971      10 PRT-ZERO-UM-WRIT REDEFINES PRT-UM-WRIT PIC ZZ,ZZ9.
972      10 FILLER                      PIC X(01).
973      10 PRT-STATED-AMT               PIC X(06).
974      10 FILLER                      PIC X(01).
975      10 PRT-COMP-DEDUCT              PIC $,$,$$.
976      10 PRT-COMP-WRIT               PIC ZZ,ZZZ.
977      10 PRT-ZERO-COMP-WRIT REDEFINES
978          PRT-COMP-WRIT PIC ZZ,ZZ9.
979      10 FILLER                      PIC X(01).
980      10 PRT-SPEC-LOSS-WRIT           PIC ZZ,ZZZ.
981      10 PRT-ZERO-SPEC-LOSS-WRIT REDEFINES
982          PRT-SPEC-LOSS-WRIT        PIC ZZ,ZZ9.
983      10 FILLER                      PIC X(03).
984      10 PRT-COLL-DEDUCT              PIC $,$,$$.
985      10 PRT-ZERO-COLL-DEDUCT REDEFINES
986          PRT-COLL-DEDUCT PIC $,$,$9.
987      10 PRT-WAIVER                   PIC X(01).
988      10 PRT-COLL-WRIT                PIC ZZ,ZZZ.
989      10 PRT-ZERO-COLL-WRIT REDEFINES
990          PRT-COLL-WRIT             PIC ZZ,ZZ9.
991      10 FILLER                      PIC X(02).
992      10 PRT-TOW-WRIT                 PIC ZZ,ZZZ.
993      10 PRT-ZERO-TOW-WRIT REDEFINES PRT-TOW-WRIT PIC ZZ,ZZ9.
994      10 FILLER                      PIC X(02).
995      10 PRT-TL-PREM                  PIC Z,ZZZ,ZZZ.
996      10 PRT-ZERO-TL-PREM REDEFINES
997          PRT-TL-PREM               PIC Z,ZZZ,ZZ9.
998      05 PRT-PREM-LINE-FL REDEFINES PRT-ACCT-LINE.
999      10 FILLER                      PIC X(14).
1000     10 PRT-FL-LIAB-WRIT             PIC ZZ,ZZZ.
1001     10 PRT-FL-ZERO-LIAB-WRIT REDEFINES
1002         PRT-FL-LIAB-WRIT          PIC ZZ,ZZ9.
1003     10 FILLER                      PIC X(06).
1004     10 PRT-FL-PIP-WRIT               PIC ZZ,ZZZ.
1005     10 PRT-FL-ZERO-PIP-WRIT REDEFINES
1006         PRT-FL-PIP-WRIT           PIC ZZ,ZZ9.
1007     10 FILLER                      PIC X(06).
1008     10 PRT-FL-ADDED-PIP-WRIT         PIC ZZ,ZZZ.

```

```

1009      10 PRT-FL-ZERO-ADDED-PIP-WRIT REDEFINES
1010      PRT-FL-ADDED-PIP-WRIT      PIC ZZ,ZZ9.
1011      10 FILLER                    PIC X(06).
1012      10 PRT-FL-MED-PAY-WRIT      PIC ZZ,ZZZ.
1013      10 PRT-FL-ZERO-MED-PAY-WRIT REDEFINES
1014      PRT-FL-MED-PAY-WRIT      PIC ZZ,ZZ9.
1015      10 FILLER                    PIC X(02).
1016      10 FILLER                    PIC X(183).
1017      05 PRT-LP-LINE  REDEFINES PRT-ACCT-LINE.
1018      10 FILLER                    PIC X(02).
1019      10 PRT-LP-POLY-NO            PIC X(15).
1020      10 FILLER                    PIC X(01).
1021      10 PRT-LP-AUTO-NO            PIC ZZZ.
1022      10 FILLER                    PIC X(04).
1023      10 PRT-LP-POLY-NAME          PIC X(38).
1024      10 FILLER                    PIC X(01).
1025      10 PRT-LP-INFO              PIC X(53).
1026      10 FILLER                    PIC X(01).
1027      10 PRT-LP-ORTD              PIC X(04).
1028      10 PRT-LP-ACCT-NO            PIC ZZZZZZZ.
1029      05 PRT-MISC-LINE REDEFINES PRT-ACCT-LINE.
1030      10 PRT-MISC-INFO            PIC X(140).
1031      05 PRT-MISC-1  REDEFINES PRT-ACCT-LINE.
1032      10 FILLER                    PIC X(43).
1033      10 PRT-MISC-1-INFO          PIC X(15).
1034      10 FILLER                    PIC X(17).
1035      05 PRT-MISC-2  REDEFINES PRT-ACCT-LINE.
1036      10 FILLER                    PIC X(30).
1037      10 PRT-MISC-2-INFO          PIC X(60).
1038      10 FILLER                    PIC X(35).
1039      05 PRT-MISC-SN2 REDEFINES PRT-ACCT-LINE.
1040      10 FILLER                    PIC X(36).
1041      10 PRT-MISC-SN2-INFO        PIC X(60).
1042      10 FILLER                    PIC X(29).
1043
1044      05 PRT-MISC-3-DEC REDEFINES PRT-ACCT-LINE.
1045      10 FILLER                    PIC X(01).
1046      10 PRT-MISC-3-INFO          PIC X(20).
1047      10 FILLER                    PIC X(113).
1048
1049      05 PRT-SR-IND-DEC REDEFINES PRT-ACCT-LINE.
1050      10 FILLER                    PIC X(01).
1051      10 PRT-SYS-RATED-IND        PIC X(124).
1052
1053
1054      05 PRT-MISC-4-COV REDEFINES PRT-ACCT-LINE.
1055      10 FILLER                    PIC X(02).
1056      10 PRT-MISC-4-INFO          PIC X(123).
1057      05 PRT-MISC-5-END REDEFINES PRT-ACCT-LINE.
1058      10 FILLER                    PIC X(02).
1059      10 PRT-MISC-5-INFO          PIC X(123).
1060      05 PRT-MISC-6-SCH REDEFINES PRT-ACCT-LINE.
1061      10 FILLER                    PIC X(08).
1062      10 PRT-MISC-6-INFO          PIC X(117).
1063      05 PRT-MISC-7-LOSS REDEFINES PRT-ACCT-LINE.
1064      10 PRT-MISC-7-INFO          PIC X(125).

```

```

1065      05 PRT-RV-MISC-1 REDEFINES PRT-ACCT-LINE.
1066      10 FILLER PIC X(03).
1067      10 PRT-RV-MISC-1-TEXT PIC X(77).
1068      05 PRT-RV-MISC-1-R REDEFINES PRT-ACCT-LINE.
1069      10 FILLER PIC X(03).
1070      10 PRT-RV-MISC-1-LEFT PIC X(65).
1071      10 PRT-RV-MISC-1-RIGHT PIC X(62).
1072      05 PRT-EXTENDED-ASSM REDEFINES PRT-ACCT-LINE.
1073      10 FILLER PIC X(03).
1074      10 PRT-EXT-ASSM-DESC PIC X(30).
1075      10 PRT-ASSM-AMOUNT PIC -,---,--9.99.
1076
1077      *-----*
1078      * WORKING STORAGE SECTION *
1079      *-----*
1080
1081      WORKING-STORAGE SECTION.
1082
1083      *-----*
1084      * THE FOLLOWING VARIABLES ARE THE PRINT LINES FOR THE REVISION. *
1085      *-----*
1086
1087      01 WS-PRT-RV-AUTO.
1088      05 FILLER PIC X(03) VALUE SPACES.
1089      05 WS-PRT-RV-AUTO-DESC-1 PIC X(07) VALUE SPACES.
1090      05 WS-PRT-HYP-1 PIC X(01).
1091      05 WS-PRT-RV-AUTO-VAR-1 PIC X(37) VALUE SPACES.
1092      05 FILLER PIC X(01) VALUE SPACES.
1093      05 WS-PRT-RV-AUTO-DESC-2 PIC X(08) VALUE SPACES.
1094      05 WS-PRT-HYP-2 PIC X(01).
1095      05 WS-PRT-RV-AUTO-VAR-2.
1096      10 WS-PRT-RV-AUTO-VAR-2-TXT PIC X(20) VALUE SPACES.
1097      05 WS-PRT-RV-AUTO-VAR-3 REDEFINES WS-PRT-RV-AUTO-VAR-2.
1098      10 WS-PRT-RV-AUTO-VAR-2-AMT PIC Z,ZZZ,ZZZ.ZZ.
1099
1100      01 WS-PRT-RV-COV.
1101      05 FILLER PIC X(03) VALUE SPACES.
1102      05 WS-PRT-RV-COV-DESC PIC X(27) VALUE SPACES.
1103      05 FILLER PIC X(01) VALUE SPACES.
1104      05 WS-PRT-RV-COV-LIMIT.
1105      10 FILLER PIC X(08) VALUE SPACES.
1106      10 WS-PRT-RV-COV-A.
1107      15 WS-PRT-RV-COV-LT PIC $,$$$,$$$$.
1108      15 FILLER PIC X(08) VALUE SPACES.
1109      10 WS-PRT-RV-COV-B REDEFINES WS-PRT-RV-COV-A.
1110      15 WS-PRT-RV-COV-LT-CHAR PIC X(17).
1111      05 WS-PRT-RV-COV-DEDUCT REDEFINES WS-PRT-RV-COV-LIMIT.
1112      10 FILLER PIC X(03).
1113      10 WS-PRT-RV-COV-DED-AMT PIC $$$9.
1114      10 FILLER PIC X(01).
1115      10 WS-PRT-RV-COV-DED-TXT PIC X(16).
1116      05 WS-PRT-RV-COV-DED-NONE REDEFINES WS-PRT-RV-COV-LIMIT.
1117      10 FILLER PIC X(05).
1118      10 WS-PRT-RV-COV-NONE-TXT PIC X(18).
1119      10 FILLER PIC X(02).
1120      05 WS-PRT-RV-COV-LIMIT-LONG REDEFINES WS-PRT-RV-COV-LIMIT.

```



```

1121         10 FILLER                PIC X(04).
1122         10 WS-PRT-RV-COV-LIMIT-LNG PIC X(21).
1123         10 FILLER                PIC X(01).
1124     05 WS-PRT-RENTAL-LONG REDEFINES WS-PRT-RV-COV-LIMIT.
1125         10 FILLER                PIC X(01).
1126         10 WS-RENTAL-REIM-PX      PIC X(22).
1127         10 FILLER                PIC X(03).
1128     05 WS-PRT-RV-COV-DOLLAR      PIC X(01).
1129     05 FILLER                    PIC X(01) VALUE SPACES.
1130     05 WS-PRT-RV-COV-PX-NET      PIC ZZZ,ZZ9.
1131     05 FILLER                    PIC X(03) VALUE SPACES.
1132     05 WS-PRT-RV-COV-DOLLAR-1    PIC X(01).
1133     05 FILLER                    PIC X(01).
1134     05 WS-PRT-RV-COV-PX          PIC ZZZ,ZZ9.
1135     05 FILLER                    PIC X(01) VALUE SPACES.
1136 *
1137     01 WS-PRT-RV-INFC-DATE.
1138         05 FILLER                PIC X(03) VALUE SPACES.
1139         05 FILLER                PIC X(23) VALUE
1140             "ENDORSEMENT EFFECTIVE: ".
1141         05 WS-PRT-RV-INFC-EFF     PIC X(10) VALUE SPACES.
1142         05 FILLER                PIC X(49) VALUE SPACES.
1143 *
1144     01 WS-PRT-RV-POLY-DATE.
1145         05 FILLER                PIC X(03) VALUE SPACES.
1146         05 FILLER                PIC X(17) VALUE
1147             "COVERAGE PERIOD: ".
1148         05 WS-PRT-RV-POLY-EFF     PIC X(10) VALUE SPACES.
1149         05 FILLER                PIC X(04) VALUE
1150             " TO ".
1151         05 WS-PRT-RV-POLY-EXP     PIC X(10) VALUE SPACES.
1152         05 FILLER                PIC X(43) VALUE SPACES.
1153 *
1154     01 WS-PRT-AUTO-COV-HEAD-1.
1155         05 FILLER                PIC X(10) VALUE SPACES.
1156         05 FILLER                PIC X(29) VALUE SPACES.
1157         05 FILLER                PIC X(18) VALUE SPACES.
1158         05 FILLER                PIC X(06) VALUE "ANNUAL".
1159         05 FILLER                PIC X(04) VALUE SPACES.
1160         05 WS-PRT-AUTO-COV-HEAD-1-TXT PIC X(10).
1161 *
1162     01 WS-PRT-AUTO-COV-HEAD-2.
1163         05 FILLER                PIC X(03) VALUE SPACES.
1164         05 FILLER                PIC X(09) VALUE "COVERAGE".
1165         05 FILLER                PIC X(29) VALUE SPACES.
1166         05 FILLER                PIC X(06) VALUE "LIMITS".
1167         05 FILLER                PIC X(09) VALUE SPACES.
1168         05 FILLER                PIC X(09) VALUE "PREMIUMS".
1169         05 FILLER                PIC X(03) VALUE SPACES.
1170         05 FILLER                PIC X(09) VALUE "PREMIUMS".
1171 *
1172     01 WS-PRT-AUTO-COV-HEAD-3.
1173         05 FILLER                PIC X(03) VALUE SPACES.
1174         05 FILLER                PIC X(09) VALUE "_____".
1175         05 FILLER                PIC X(29) VALUE SPACES.
1176         05 FILLER                PIC X(06) VALUE "_____".

```

```

1177      05 FILLER          PIC X(09) VALUE SPACES.
1178      05 FILLER          PIC X(09) VALUE "_____".
1179      05 FILLER          PIC X(03) VALUE SPACES.
1180      05 FILLER          PIC X(09) VALUE "_____".
1181  *
1182  01 WS-PRT-AUTO-LINE.
1183      05 FILLER          PIC X(65) VALUE
1184      "_____
1185  -  "_____".
1186      05 FILLER          PIC X(15) VALUE
1187      "_____".
1188  *
1189  01 WS-PRT-AUTO-LINE-2.
1190      05 FILLER          PIC X(65) VALUE
1191      "_____
1192  -  "_____".
1193      05 FILLER          PIC X(60) VALUE
1194      "_____
1195  -  "_____".
1196  *
1197  01 WS-PRT-RV-VEHL-PX-LINE.
1198      05 FILLER          PIC X(03) VALUE SPACES.
1199      05 FILLER          PIC X(30) VALUE
1200      "FOR THIS ENDORSEMENT THERE IS ".
1201      05 WS-PRT-RV-VEHL-TEXT  PIC X(28).
1202      05 FILLER          PIC X(01).
1203      05 WS-PRT-RV-VEHL-PX-DOL  PIC X(01).
1204      05 WS-PRT-RV-VEHL-PX-INCL.
1205          10 WS-PRT-RV-VEHL-AMT.
1206              15 WS-PRT-RV-VEHL-PX  PIC ZZZ,ZZZ,ZZ9.99.
1207          10 WS-PRT-RV-VEHL-INCLUDE REDEFINES WS-PRT-RV-VEHL-AMT.
1208              15 WS-PRT-RV-VEHL-INCL  PIC X(14).
1209  *
1210  01 WS-PRT-RV-VEHL-TEXT-RETURN  PIC X(28) VALUE
1211      "A RETURN PREMIUM DUE OF".
1212  *
1213  01 WS-PRT-RV-VEHL-TEXT-ADD  PIC X(28) VALUE
1214      "AN ADDITIONAL PREMIUM DUE OF".
1215  *
1216  01 WS-PRT-RV-AUTO-DEL  PIC X(80) VALUE
1217      "THE FOLLOWING COVERED AUTO IS DELETED :".
1218  *
1219  01 WS-PRT-RV-AUTO-ADD  PIC X(80) VALUE
1220      "THE FOLLOWING COVERED AUTO IS ADDED :".
1221  *
1222  01 WS-PRT-RV-AUTO-AMENDED  PIC X(80) VALUE
1223      "THE COVERAGE HAS BEEN AMENDED ON THE FOLLOWING VEHICLE :".
1224  *
1225  01 WS-PRT-RV-AUTO-END.
1226      05 FILLER          PIC X(30) VALUE SPACES.
1227      05 FILLER          PIC X(14) VALUE "ENDORSEMENT #".
1228      05 WS-PRT-RV-AUTO-END-NO PIC ZZZZ.
1229      05 FILLER          PIC X(30) VALUE SPACES.
1230  *
1231  01 WS-PRT-RV-HEADING.
1232      05 FILLER          PIC X(32) VALUE SPACES.

```

```

1233      05 FILLER          PIC X(14) VALUE "VEHICLE CHANGE".
1234      05 FILLER          PIC X(32) VALUE SPACES.
1235  *
1236      01 WS-PRT-RV-AUTO-POLY.
1237          05 FILLER          PIC X(52) VALUE
1238              " Attached to and forming part of Policy Number:".
1239          05 WS-PRT-RV-POLY-NO PIC X(15).
1240          05 FILLER          PIC X(01) VALUE SPACES.
1241          05 FILLER          PIC X(15) VALUE
1242              "Account Number:".
1243          05 WS-PRT-RV-ACCT-NO PIC ZZZZZZZ.
1244  *
1245      01 WS-PRT-RV-TERMS.
1246          05 FILLER          PIC X(62) VALUE
1247              "All other terms and conditions of the policy
1248  -      " remain unchanged.".
1249  *
1250      01 WS-PRT-RV-AGENT.
1251          05 FILLER          PIC X(57) VALUE
1252              " _____ Agent".
1253  *
1254      01 WS-PRT-RV-PAGE.
1255          05 FILLER          PIC X(05) VALUE "PAGE ".
1256          05 WS-PRT-RV-NO     PIC ZZZ.
1257          05 FILLER          PIC X(04) VALUE " OF ".
1258          05 WS-PRT-RV-NO-OF  PIC ZZZ.
1259  *
1260  * -----*
1261  * THE FOLLOWING VARIABLES ARE FOR THE PRINT NOTICE OF AUTOMOBILE  *
1262  * INSURANCE LINE. *
1263  * -----*
1264  *
1265      01 WS-NOTICE-HEAD-1.
1266          05 FILLER          PIC X(25) VALUE SPACES.
1267          05 FILLER          PIC X(30) VALUE
1268              "NOTICE OF AUTOMOBILE INSURANCE".
1269          05 FILLER          PIC X(25) VALUE SPACES.
1270  *
1271      01 WS-NOTICE-HEAD-2.
1272          05 FILLER          PIC X(05) VALUE SPACES.
1273          05 FILLER          PIC X(24) VALUE
1274              "NAMED INSURED & ADDRESS:".
1275          05 FILLER          PIC X(22) VALUE SPACES.
1276          05 FILLER          PIC X(44) VALUE
1277              "COMPANY: FLORISTS' MUTUAL INSURANCE COMPANY".
1278
1279      01 WS-NOTICE-HEAD-SN.
1280          05 FILLER          PIC X(05) VALUE SPACES.
1281          05 FILLER          PIC X(24) VALUE
1282              "NAMED INSURED & ADDRESS:".
1283          05 FILLER          PIC X(19) VALUE SPACES.
1284          05 FILLER          PIC X(47) VALUE
1285              "COMPANY: State National Insurance Company, Inc.".
1286  *
1287      01 WS-NOTICE-HEAD-3.
1288          05 FILLER          PIC X(05) VALUE SPACES.

```

```

1289      05 WS-NOTICE-LP-COMPANY PIC X(50) VALUE SPACES.
1290      05 HEAD-3-FM-LAYOUT.
1291      10 FILLER PIC X(05) VALUE SPACES.
1292      10 WS-NOTICE-FM-COMPANY PIC X(30).
1293      05 HEAD-3-SN-LAYOUT REDEFINES HEAD-3-FM-LAYOUT.
1294      10 FILLER PIC X(02).
1295      10 WS-NOTICE-SN-COMPANY PIC X(33).
1296      *
1297      01 WS-NOTICE-HEAD-7.
1298      05 FILLER PIC X(05) VALUE SPACES.
1299      05 FILLER PIC X(11) VALUE
1300      "POLICY NO:".
1301      05 WS-NOTICE-POLY-NO PIC X(15) VALUE SPACES.
1302      05 FILLER PIC X(22) VALUE SPACES.
1303      05 FILLER PIC X(23) VALUE
1304      "ENDORSEMENT EFFECTIVE:".
1305      05 WS-NOTICE-INFC-EFF PIC X(10) VALUE SPACES.
1306      *
1307      01 WS-NOTICE-HEAD-8.
1308      05 FILLER PIC X(05) VALUE SPACES.
1309      05 FILLER PIC X(26) VALUE
1310      "EFFECTIVE DATE OF POLICY:".
1311      05 WS-NOTICE-POLY-EFF PIC X(10) VALUE SPACES.
1312      05 FILLER PIC X(12) VALUE SPACES.
1313      05 FILLER PIC X(26) VALUE
1314      "EXPIRATION DATE OF POLICY:".
1315      05 WS-NOTICE-POLY-EXP PIC X(10) VALUE SPACES.
1316      *
1317      01 WS-NOTICE-HEAD-8-SN.
1318      05 FILLER PIC X(05) VALUE SPACES.
1319      05 FILLER PIC X(26) VALUE
1320      "EFFECTIVE DATE OF POLICY:".
1321      05 WS-NOTICE-POLY-EFF-SN PIC X(10) VALUE SPACES.
1322      05 FILLER PIC X(12) VALUE SPACES.
1323      05 FILLER PIC X(26) VALUE
1324      "EXPIRATION DATE OF POLICY:".
1325      05 WS-NOTICE-POLY-EXP-SN PIC X(10) VALUE SPACES.
1326      *
1327      01 WS-HOLD-COMPANY PIC X(27).
1328      01 WS-HOLD-POBOX PIC X(19).
1329      01 WS-HOLD-CITY PIC X(31).
1330      *
1331      01 WS-NOTICE-HEAD-9.
1332      05 FILLER PIC X(05) VALUE SPACES.
1333      05 FILLER PIC X(12) VALUE
1334      "LOSS PAYEE:".
1335      05 WS-NOTICE-LP-NAME PIC X(60) VALUE SPACES.
1336      05 FILLER PIC X(03) VALUE SPACES.
1337      *
1338      01 WS-NOTICE-HEAD-10.
1339      05 FILLER PIC X(17) VALUE SPACES.
1340      05 WS-NOTICE-LP-STREET1 PIC X(60) VALUE SPACES.
1341      05 FILLER PIC X(03) VALUE SPACES.
1342      *
1343      01 WS-NOTICE-HEAD-11.
1344      05 FILLER PIC X(17) VALUE SPACES.

```

```

1345      05 WS-NOTICE-LP-STREET2 PIC X(60) VALUE SPACES.
1346      05 FILLER                PIC X(03) VALUE SPACES.
1347  *
1348      01 WS-NOTICE-HEAD-12.
1349          05 FILLER                PIC X(17) VALUE SPACES.
1350          05 WS-NOTICE-LP-CITY-ST PIC X(60) VALUE SPACES.
1351          05 FILLER                PIC X(03) VALUE SPACES.
1352  *
1353      01 WS-NOTICE-HEAD-13.
1354          05 FILLER                PIC X(52) VALUE SPACES.
1355          05 FILLER                PIC X(33) VALUE
1356          "_____".
1357  *
1358      01 WS-NOTICE-HEAD-13-B.
1359          05 FILLER                PIC X(55) VALUE SPACES.
1360          05 FILLER                PIC X(27) VALUE
1361          "(Authorized Representative)".
1362  *
1363      01 WS-NOTICE-HEAD-14.
1364          05 FILLER                PIC X(10) VALUE SPACES.
1365          05 FILLER                PIC X(09) VALUE
1366          "COVERAGES".
1367          05 FILLER                PIC X(44) VALUE SPACES.
1368          05 FILLER                PIC X(18) VALUE
1369          "LIMIT OF LIABILITY".
1370  *
1371  *****
1372      01 WS-NOTICE-HEAD-15.
1373          05 FILLER                PIC X(05) VALUE SPACES.
1374          05 WS-NOTICE-COV-TYPE PIC X(35) VALUE SPACES.
1375          05 FILLER                PIC X(02) VALUE SPACES.
1376          05 WS-NOTICE-LIMIT.
1377              10 FILLER          PIC X(22).
1378              10 WS-NOTICE-LIMIT-A.
1379                  15 WS-NOTICE-LIMIT-AMT PIC $$$$$$.
1380              10 WS-NOTICE-LIMIT-B REDEFINES WS-NOTICE-LIMIT-A.
1381                  15 WS-NOTICE-LIMIT-AMT-CHAR PIC X(09).
1382              10 FILLER          PIC X(01).
1383              10 WS-NOTICE-LIMIT-EACH PIC X(18).
1384          05 WS-NOTICE-ACV REDEFINES WS-NOTICE-LIMIT.
1385              10 WS-NOTICE-ACV-TXT PIC X(47).
1386  *
1387          05 WS-BI-AMT-NUMERIC REDEFINES WS-NOTICE-LIMIT.
1388              10 WS-BI-AMT-HEAD PIC $$$,$$$,$$$$.
1389              10 WS-SLASH-1-HEAD PIC X(01).
1390              10 WS-BI-AMT-2-HEAD PIC $$$,$$$,$$$$.
1391              10 WS-SLASH-2-HEAD PIC X(01).
1392              10 WS-PD-AMT-HEAD PIC $$$,$$$,$$$$.
1393          05 WS-NOTICE-DEDUCT REDEFINES WS-NOTICE-LIMIT.
1394              10 FILLER          PIC X(22).
1395              10 WS-NOTICE-DEDUCT-A.
1396                  15 WS-NOTICE-DEDUCT-AMT PIC $$$$$$.
1397              10 WS-NOTICE-DEDUCT-B REDEFINES WS-NOTICE-DEDUCT-A.
1398                  15 WS-ZERO-NOT-DEDUCT-AMT PIC $$$9.
1399              10 FILLER          PIC X(01).
1400              10 WS-NOTICE-DEDUCT-COV PIC X(21).

```

```

1401 *
1402 01 WS-NOTICE-HEAD-16.
1403     05 FILLER          PIC X(05) VALUE SPACES.
1404     05 FILLER          PIC X(76) VALUE
1405     "THIS NOTICE OF AUTOMOBILE INSURANCE IS APPLICABLE AS
1406 -   " MARKED BELOW.".
1407 *
1408 01 WS-NOTICE-HEAD-17.
1409     05 FILLER          PIC X(05) VALUE SPACES.
1410     05 FILLER          PIC X(01) VALUE "[".
1411     05 WS-NOTICE-HEAD-17-X PIC X(01).
1412     05 FILLER          PIC X(01) VALUE "]".
1413     05 FILLER          PIC X(02) VALUE SPACES.
1414     05 FILLER          PIC X(60) VALUE
1415     "This certifies that the policy listed above
1416 -   " has been issued".
1417     05 FILLER          PIC X(60) VALUE
1418     "to the named insured and is in force at this time".
1419 *
1420 01 WS-NOTICE-HEAD-18.
1421     05 FILLER          PIC X(10) VALUE SPACES.
1422     05 FILLER          PIC X(58) VALUE
1423     "for the auto(s) described. The insurance afforded by the".
1424     05 FILLER          PIC X(60) VALUE
1425     "policy described herein is subject to all the terms,".
1426 *
1427 01 WS-NOTICE-HEAD-19.
1428     05 FILLER          PIC X(10) VALUE SPACES.
1429     05 FILLER          PIC X(70) VALUE
1430     "exclusions and conditions of such policy.".
1431 *
1432 01 WS-NOTICE-HEAD-20.
1433     05 FILLER          PIC X(10) VALUE SPACES.
1434     05 FILLER          PIC X(55) VALUE
1435     "Except for towing, all physical damage loss is payable".
1436     05 FILLER          PIC X(60) VALUE
1437     "to the Named Insured and the Loss Payee named above".
1438 *
1439 01 WS-NOTICE-HEAD-21.
1440     05 FILLER          PIC X(10) VALUE SPACES.
1441     05 FILLER          PIC X(70) VALUE
1442     "as interest may appear at the time of loss.".
1443 *
1444 01 WS-NOTICE-HEAD-22.
1445     05 FILLER          PIC X(05) VALUE SPACES.
1446     05 FILLER          PIC X(01) VALUE "[".
1447     05 WS-NOTICE-HEAD-22-X PIC X(01).
1448     05 FILLER          PIC X(01) VALUE "]".
1449     05 FILLER          PIC X(02) VALUE SPACES.
1450     05 FILLER          PIC X(54) VALUE
1451     "This is to notify that the policy of insurance listed".
1452     05 FILLER          PIC X(65) VALUE
1453     "above has been changed by the insured to delete the Loss".
1454 *
1455 01 WS-NOTICE-HEAD-23.
1456     05 FILLER          PIC X(10) VALUE SPACES.

```



```

1457      05 FILLER          PIC X(70) VALUE
1458      "Payee named above effective _____.".
1459  *
1460  01 WS-NOTICE-HEAD-24.
1461      05 FILLER          PIC X(45) VALUE SPACES.
1462      05 WS-NOTICE-HEAD-24-DATE PIC X(10).
1463  *
1464  01 WS-NOTICE-MI1.
1465      05 FILLER          PIC X(51) VALUE
1466      "This policy is exempt from the filing requirements ".
1467      05 FILLER          PIC X(43) VALUE
1468      "of section 2236 of the insurance code 1956,".
1469  01 WS-NOTICE-MI2.
1470      05 FILLER          PIC X(54) VALUE
1471      "1956 PA 218, MCL 500.2236.".
1472  *
1473  * -----*
1474  *  ESCAPE CODE FOR PRINTING THE PLAIN PAPER.
1475  * -----*
1476  *
1477  01 WS-REG-FORM-ESC.
1478      05 WS-SPEC-FORM-ESC-2      PIC X
1479      VALUE X"1B".
1480      05 WS-SPEC-FORM-ESC-2      PIC X(07)
1481      VALUE "&f99y4X".
1482  *
1483  * -----*
1484  *  GENERATIC ESCAPE CODE LINE
1485  * -----*
1486  *
1487  01 ESC          PIC X VALUE X"1B".
1488  *
1489  01 WS-PRINT-MACRO-LINE.
1490      05 WS-PRINT-ESC          PIC X(01).
1491      05 WS-PRINT-MACRO        PIC X(09).
1492  *
1493  01 IDREQFNT.
1494      05 IDFN-ESC1          PIC X    VALUE X"1B".
1495      05 IDFN-1            PIC X(04) VALUE "&100".
1496      05 IDFN-ESC2          PIC X    VALUE X"1B".
1497      05 IDFN-2            PIC X(03) VALUE "(8U".
1498      05 IDFN-ESC3          PIC X    VALUE X"1B".
1499      05 IDFN-3            PIC X(16)
1500      VALUE "(s0p06h24v0s3b3T".
1501  *
1502  * -----*
1503  *  THE FOLLOWING CODE WILL SET THE LASER PRINT UP TO PRINT IN "NORMAL"
1504  *  10 CHARACTERS PER INCH PRINT.
1505  * -----*
1506  *
1507  01 WS-COURIER.
1508      05 COUR-ESC1          PIC X    VALUE X"1B".
1509      05 COUR-1            PIC X(04) VALUE "&100".
1510      05 COUR-ESC2          PIC X    VALUE X"1B".
1511      05 COUR-2            PIC X(03) VALUE "(8U".
1512      05 COUR-ESC3          PIC X    VALUE X"1B".

```

```

1513         05 COUR-3          PIC X(16)
1514         VALUE "(s0p10h12v0s0b3T".
1515
1516 *-----*
1517 * THE FOLLOWING CODE WILL SET THE LASER PRINT UP TO PRINT 12 CHARACTERS *
1518 * PER INCH PRINT. *
1519 *-----*
1520 *
1521 01 WS-12-CHAR.
1522     05 WS-ESC1          PIC X    VALUE X"1B".
1523     05 WS-1            PIC X(04) VALUE "&10O".
1524     05 WS-ESC2          PIC X    VALUE X"1B".
1525     05 WS-2            PIC X(03) VALUE "(8U".
1526     05 WS-ESC3          PIC X    VALUE X"1B".
1527     05 WS-3            PIC X(25)
1528         VALUE "(s0p12.0h10.0v0s0b3T".
1529
1530 *-----*
1531 * THE FOLLOWING CODE WILL SET THE LASER PRINT UP TO PRINT 12 characters *
1532 * per inch IN BOLD PRINT. *
1533 *-----*
1534 *
1535 01 WS-12-BOLD.
1536     05 WS-BOLD-ESC1     PIC X    VALUE X"1B".
1537     05 WS-BOLD-1        PIC X(04) VALUE "&10O".
1538     05 WS-BOLD-ESC2     PIC X    VALUE X"1B".
1539     05 WS-BOLD-2        PIC X(03) VALUE "(8U".
1540     05 WS-BOLD-ESC3     PIC X    VALUE X"1B".
1541     05 WS-BOLD-3        PIC X(25)
1542         VALUE "(s0p12.0h10.0v0s3b3T".
1543     05 WS-BOLD-ESC4     PIC X    VALUE X"1B".
1544     05 WS-BOLD-4        PIC X(04) VALUE "&10O".
1545
1546 *-----*
1547 * THE FOLLOWING CODE WILL SET THE LASER PRINT UP TO PRINT 10 characters *
1548 * per inch IN BOLD PRINT. *
1549 *-----*
1550 *
1551 01 WS-BOLD.
1552     05 BOLD-ESC1        PIC X    VALUE X"1B".
1553     05 BOLD-1           PIC X(04) VALUE "&10O".
1554     05 BOLD-ESC2        PIC X    VALUE X"1B".
1555     05 BOLD-2           PIC X(03) VALUE "(8U".
1556     05 BOLD-ESC3        PIC X    VALUE X"1B".
1557     05 BOLD-3           PIC X(25)
1558         VALUE "(s0p10.0h12.0v0s3b3T".
1559     05 BOLD-ESC4        PIC X    VALUE X"1B".
1560     05 BOLD-4           PIC X(04) VALUE "&10O".
1561
1562 *-----*
1563 * THE FOLLOWING CODE WILL SET THE LASER PRINT UP TO PRINT IN COMPRESSED *
1564 * (16.6 characters per inch) PRINT. *
1565 *-----*
1566 *
1567 01 WS-LINEPTR.
1568     05 LINE-ESC1        PIC X    VALUE X"1B".

```

```

1569      05 LINE-1          PIC X(04) VALUE "&100".
1570      05 LINE-ESC2      PIC X      VALUE X"1B".
1571      05 LINE-2          PIC X(03) VALUE "(8U".
1572      05 LINE-ESC3      PIC X      VALUE X"1B".
1573      05 LINE-3          PIC X(20)
1574          VALUE "(s0p16.66h8.5v0s0b0T".
1575      *
1576      * -----*
1577      * THIS ESCAPE SEQUENCE SETS THE PRINTING OF LINES PER INCH TO 8. *
1578      * -----*
1579      *
1580      01 LINEFEED-8.
1581          05 FEED-ESC-8   PIC X(01).
1582          05 FEED-LINE8   PIC X(05) VALUE "&l12D".
1583          05 COL-ESC      PIC X(01).
1584          05 COL-POS      PIC X(06) VALUE "&a+13C".
1585      *
1586      * -----*
1587      * THIS ESCAPE SEQUENCE SETS THE PRINTING OF LINES PER INCH TO 6. *
1588      * -----*
1589      *
1590      01 LINEFEED-6.
1591          05 FEED-ESC-6   PIC X(01).
1592          05 FEED-LINE6   PIC X(04) VALUE "&l6D".
1593      *
1594      * -----*
1595      * COUNTING VARIABLES. *
1596      * -----*
1597      *
1598      01 WS-LINE-COUNT     PIC 9(02) VALUE 0.
1599      01 WS-BLANK-LINE    PIC 9(02) VALUE 0.
1600      01 WS-ENDORSEMENT-COUNT PIC 9(02) VALUE 0.
1601      01 WS-STATE-COUNT   PIC 9(02) VALUE 0.
1602      01 WS-PG2-LINECNT   PIC 9(02) VALUE 0.
1603      01 WS-ASSM-CNT      PIC 9(02) VALUE 0.
1604      *
1605      * -----*
1606      * MISC VARIABLES. *
1607      * -----*
1608      *
1609      01 WS-NON-REMAINDER PIC S9(03) VALUE 0.
1610      01 WS-REMAINDER     PIC S9V999 VALUE 0.
1611      01 WS-POLY-NO       PIC X(15) VALUE SPACES.
1612      01 WS-ERR-MSG       PIC X(80) VALUE SPACES.
1613      01 WS-DISP-NO       PIC 9(07) VALUE 0.
1614      01 WS-PRT-ENDFORM   PIC X(27) VALUE SPACES.
1615      01 WS-LIAB-COST      PIC S9(11) VALUE 0.
1616      01 WS-LIAB-COST-NET PIC S9(11) VALUE 0.
1617      01 WS-DISPLAY-NUMBER PIC S9(07) VALUE 0.
1618      01 WS-CITY          PIC X(20) VALUE SPACES.
1619      01 WS-ZIP           PIC X(09) VALUE SPACES.
1620      01 WS-STATE         PIC X(02) VALUE SPACES.
1621      01 WS-CNT-IL-LOC    PIC 9(03) VALUE 0.
1622      01 WS-CNT-WI-LOC    PIC 9(03) VALUE 0.
1623      01 WS-CNT-VT-LOC    PIC 9(03) VALUE 0.
1624      01 WS-CNT-LA-LOC    PIC 9(03) VALUE 0.

```

1625	01	WS-TERR-CD	PIC X(05)	VALUE SPACES.
1626	01	ws-grg-sts	PIC 9(03)	VALUE 0.
1627				
1628	*			
1629	01	WS-POSITION-COUNT	PIC 9(02)	VALUE 0.
1630	01	WS-POSITION-COUNT-HOLD	PIC 9(02)	VALUE 0.
1631	01	WS-TEST-POSITION	PIC X(01)	VALUE LOW-VALUES.
1632	01	WS-BI-AMT	PIC X(09)	VALUE SPACES.
1633	01	WS-BI-AMT-2	PIC X(09)	VALUE SPACES.
1634	01	WS-PD-AMT	PIC X(09)	VALUE SPACES.
1635	01	WS-SLASH	PIC X(01)	VALUE "/".
1636	01	WS-PD-AMT-LENGTH	PIC 9(02)	VALUE 0.
1637	01	WS-BI-AMT-LENGTH	PIC 9(02)	VALUE 0.
1638	01	WS-COMMA-POSITION-COUNT	PIC 9(02)	VALUE 0.
1639	01	WS-COMMA	PIC X(01)	VALUE ",".
1640	01	WS-DOLLAR-SIGN	PIC X(01)	VALUE "\$".
1641	01	WS-PD-OUT	PIC X(12)	VALUE SPACES.
1642	01	WS-PD-OUT-HOLD	PIC X(12)	VALUE SPACES.
1643	01	WS-PD-OUT-FINAL	PIC X(12)	VALUE SPACES.
1644	01	WS-BI-OUT	PIC X(12)	VALUE SPACES.
1645	01	WS-BI-OUT-HOLD	PIC X(12)	VALUE SPACES.
1646	01	WS-BI-OUT-FINAL	PIC X(12)	VALUE SPACES.
1647	01	WS-BI-OUT-2	PIC X(12)	VALUE SPACES.
1648	01	WS-BI-OUT-HOLD-2	PIC X(12)	VALUE SPACES.
1649	01	WS-BI-OUT-2-FINAL	PIC X(12)	VALUE SPACES.
1650	*			
1651	01	WS-ERR-MSG-TABLE.		
1652	05	WS-ERR-MSG-TBL	PIC X(80)	VALUE SPACES
1653		OCCURS	100 TIMES.	
1654	*			
1655	01	WS-WHAT-SUB	PIC S9(7)	VALUE 0.
1656	01	WS-NEXT-SUB	PIC S9(7)	VALUE 0.
1657	01	WS-KY-SUB	PIC S9(7)	VALUE 0.
1658	01	WS-ASSM-COUNTER	PIC S9(7)	VALUE 0.
1659	01	WS-FOOTER	PIC S9(7)	VALUE 0.
1660	01	WS-REDUCT	PIC S9(7)	VALUE 0.
1661	*			
1662	01	WS-HCOLL-DEDUCT	PIC X(09)	VALUE SPACES.
1663	01	WS-HCOLL-LIMIT	PIC X(09)	VALUE SPACES.
1664	01	WS-DEDUCT-LEN	PIC 9(02)	VALUE 0.
1665	01	WS-LIMIT-LEN	PIC 9(02)	VALUE 0.
1666	01	WS-LOC-STATE	PIC X(02)	VALUE SPACES.
1667	01	WS-MCTB-EDIT	PIC X(01)	VALUE SPACES.
1668	01	WS-WA-HOLD-LIMIT	PIC X(11)	VALUE SPACES.
1669	01	WS-WA-HOLD-PX	PIC X(12)	VALUE SPACES.
1670	01	WS-STATE-VAR22	PIC X(19)	VALUE SPACES.
1671	01	WS-STATE-VAR7	PIC X(07)	VALUE SPACES.
1672	01	WS-STATE-VAR9	PIC X(09)	VALUE SPACES.
1673	01	WS-COVG-COUNT	PIC 9(02)	VALUE 0.
1674	01	WS-COVG-STATE	PIC X(02)	VALUE SPACES.
1675	01	WS-AUCV-CODE	PIC X(02)	VALUE SPACES.
1676	01	WS-AUCV-TYPE	PIC X(02)	VALUE SPACES.
1677	01	WS-COVG-TEXT	PIC X(22)	VALUE SPACES.
1678	01	WS-IL-COVGTEXT	PIC X(22)	VALUE SPACES.
1679	01	WS-VT-COVGTEXT	PIC X(22)	VALUE SPACES.
1680	01	WS-VT-COVTXT	PIC X(09)	VALUE "Uninsured".

```

1681 01 WS-WI-COVGTEXT      PIC X(22)  VALUE SPACES.
1682 01 WS-LA-COVGTEXT      PIC X(22)  VALUE SPACES.
1683 01 WS-TXSD-CLASS       pic x(05)   value spaces.
1684 01 WS-TXDS-LIMIT       pic 9(05)   value 0.
1685 01 WS-TXDS-PX         pic 9(03)   value 0.
1686
1687 01 WS-HA-IND            PIC X(01) VALUE SPACES.
1688 01 WS-NO-IND            PIC X(01) VALUE SPACES.
1689 01 WS-UM-IND            PIC X(01) VALUE SPACES.
1690 01 WS-UI-IND            PIC X(01) VALUE SPACES.
1691 01 WS-MED-IND           PIC X(01) VALUE SPACES.
1692 01 WS-HAMED-IND        PIC X(01) VALUE SPACES.
1693 01 WS-NOMED-IND        PIC X(01) VALUE SPACES.
1694 01 WS-MED-HOLD         PIC X(01) VALUE SPACES.
1695 01 WS-HOLD-EDITION     PIC X(25) VALUE SPACES.
1696 01 WS-SN-111-ISSUED    PIC X(55)
1697     VALUE " State National Insurance Company, Inc. ISSUED TO:".
1698
1699 *-----*
1700 *                SWITCHES                *
1701 *-----*
1702 *
1703 01 WS-SNGL-ST          PIC X(01) VALUE "N".
1704 01 WS-ALRDY-DN         PIC X(01) VALUE "N".
1705 01 WS-EOC-LOC          PIC X(01) VALUE "N".
1706 01 WS-EOC-ASSM-1      PIC X(01) VALUE "N".
1707 01 WS-EOC-ASSM-2      PIC X(01) VALUE "N".
1708 01 WS-EOC-DRIVING     PIC X(01) VALUE "N".
1709 01 WS-EOC-ATTACH-CURS PIC X(01) VALUE "N".
1710 01 WS-EOC-AUTO        PIC X(01) VALUE "N".
1711 01 WS-EOC-AUTO-PREM   PIC X(01) VALUE "N".
1712 01 WS-EOC-VINT-SW     PIC X(01) VALUE "N".
1713 01 WS-EOC-ENDORSE     PIC X(01) VALUE "N".
1714 01 WS-EOC-RV-CURSOR   PIC X(01) VALUE "N".
1715 01 WS-EOC-OTHER-COV   PIC X(01) VALUE "N".
1716 01 WS-EOC-LOC-STATE   PIC X(01) VALUE "N".
1717 01 WS-HOLD-STATE      PIC X(02) VALUE SPACES.
1718 01 WS-IL-LOC          PIC X(01) VALUE "N".
1719 01 WS-WI-LOC          PIC X(01) VALUE "N".
1720 01 WS-VT-LOC          PIC X(01) VALUE "N".
1721 01 WS-LA-LOC          PIC X(01) VALUE "N".
1722 01 WS-WI-IL-LOC       PIC X(01) VALUE "N".
1723 01 WS-VT-IL-LOC       PIC X(01) VALUE "N".
1724 01 WS-VT-WI-LOC       PIC X(01) VALUE "N".
1725 01 WS-HOLD-PPRT-INFC-GB-NO PIC S9(07) COMP-3  VALUE 0.
1726 01 WS-VAR-PIP-EXIST    PIC X(01) VALUE "N".
1727 01 WS-VAR-NY-PIP-EXIST PIC X(01) VALUE "N".
1728 01 WS-NYAPIP-EXIST     PIC X(01) VALUE "N".
1729 01 WS-NYAPDTH-EXIST    PIC X(01) VALUE "N".
1730 01 WS-VAR-COMP-EXIST    PIC X(02) VALUE "N".
1731 01 WS-VAR-SPEC-LOSS-EXIST PIC X(02) VALUE "N".
1732 01 WS-VAR-COLL-EXIST    PIC X(02) VALUE "N".
1733 01 WS-GLASS-EXIST-L     PIC X(02) VALUE "N".
1734 01 WS-GLASS-EXIST-M     PIC X(02) VALUE "N".
1735 01 WS-VAR-TOW-EXIST     PIC X(02) VALUE "N".
1736 01 WS-VAR-ADPIP-EXIST   PIC X(02) VALUE "N".

```



```

1737 01 WS-VAR-PPI-EXIST      PIC X(02) VALUE "N".
1738 01 WS-VAR-BUY-EXIST      PIC X(02) VALUE "N".
1739 01 WS-VAR-MEDEXP-EXIST   PIC X(02) VALUE "N".
1740 01 WS-VAR-ILB-EXIST     PIC X(02) VALUE "N".
1741 01 WS-NY-EXIST          PIC X(02) VALUE "N".
1742 01 WS-VA-EXIST          PIC X(02) VALUE "N".
1743 01 WS-MI-EXIST          PIC X(02) VALUE "N".
1744 01 WS-PA-EXIST          PIC X(02) VALUE "N".
1745 01 WS-NC-EXIST          PIC X(02) VALUE "N".
1746 01 WS-KY-EXIST          PIC X(02) VALUE "N".
1747 01 WS-WA-EXIST          PIC X(02) VALUE "N".
1748 01 WS-NJ-EXIST          PIC X(02) VALUE "N".
1749 01 WS-WA-UIPD-EXIST     PIC X(02) VALUE "N".
1750 01 WS-MULTISTATE         PIC X(02) VALUE "N".
1751 01 WS-SSL-EXIST          PIC X(02) VALUE "N".
1752 01 WS-PRT-2A            PIC X(02) VALUE "N".
1753 01 WS-NY-SUM-EXIST       PIC X(02) VALUE "N".
1754 01 WS-STUM-EXIST         PIC X(02) VALUE "N".
1755 01 WS-UMPD-LIMIT1        PIC X(02) VALUE "N".
1756 01 WS-PAGE-3A           PIC X(02) VALUE "N".
1757 01 EOF-UMUI-LOC         PIC 9 VALUE ZERO.
1758      88 END-OF-UMUI-LOC   VALUE 1.
1759 01 EOF-TXSD              PIC 9 VALUE ZERO.
1760      88 END-OF-TXSD        VALUE 1.
1761
1762 *
1763 * -----*
1764 *                TABLES                *
1765 * -----*
1766 *
1767 01 WS-ENDORSEMENT-TABLE.
1768     05 WS-END-1           PIC X(27) VALUE SPACES
1769         OCCURS 200 TIMES.
1770     05 WS-END-2           PIC X(27) VALUE SPACES
1771         OCCURS 200 TIMES.
1772     05 WS-END-3           PIC X(27) VALUE SPACES
1773         OCCURS 200 TIMES.
1774     05 WS-END-4           PIC X(27) VALUE SPACES
1775         OCCURS 200 TIMES.
1776 *
1777 * -----*
1778 * THIS TABLE WILL BE USED TO PRINT THE INFORMATION NEEDED ON THE      *
1779 * SCHEDULE OF AUTOMOBILES. THIS IS INFORMATION IS THE VEHICLE INFO.    *
1780 * -----*
1781 *
1782 01 WS-SCHEDULE-AUTO-TABLE.
1783     05 WS-TBL-ID-NO        PIC 9(04)
1784         VALUE 0 OCCURS 1000 TIMES.
1785     05 WS-TBL-SERIAL-NO    PIC X(20)
1786         VALUE SPACES OCCURS 1000 TIMES.
1787     05 WS-TBL-VHDS-AU-NO   PIC 9(07)
1788         VALUE 0 OCCURS 1000 TIMES.
1789     05 WS-TBL-YEAR         PIC 9(04)
1790         VALUE 0 OCCURS 1000 TIMES.
1791     05 WS-TBL-MAKE         PIC X(20)
1792         VALUE SPACES OCCURS 1000 TIMES.

```


1793	05 WS-TBL-MODEL	PIC X(20)	
1794	VALUE SPACES OCCURS 1000 TIMES.		
1795	05 WS-TBL-POP-DENSITY	PIC X(04)	
1796	VALUE SPACES OCCURS 1000 TIMES.		
1797	05 WS-TBL-COST	PIC S9(07) COMP-3	
1798	VALUE 0 OCCURS 1000 TIMES.		
1799	05 WS-TBL-LOSS-PAYEE	PIC X(03)	
1800	VALUE SPACES OCCURS 1000 TIMES.		
1801	05 WS-TBL-DRIVER-REC	PIC X(12)	
1802	VALUE SPACES OCCURS 1000 TIMES.		
1803	05 WS-TBL-MILEAGE	PIC X(07)	
1804	VALUE SPACES OCCURS 1000 TIMES.		
1805	05 WS-TBL-USE	PIC X(02)	
1806	VALUE SPACES OCCURS 1000 TIMES.		
1807	05 WS-TBL-WEIGHT	PIC X(04)	
1808	VALUE SPACES OCCURS 1000 TIMES.		
1809	05 WS-TBL-CITY	PIC X(20)	
1810	VALUE SPACES OCCURS 1000 TIMES.		
1811	05 WS-TBL-STATE	PIC X(02)	
1812	VALUE SPACES OCCURS 1000 TIMES.		
1813	05 WS-TBL-CLASS-CODE	PIC X(06)	
1814	VALUE SPACES OCCURS 1000 TIMES.		
1815	05 WS-TBL-CLASS	PIC X(02)	
1816	VALUE SPACES OCCURS 1000 TIMES.		
1817	05 WS-TBL-USAGE-TYPE	PIC X(02)	
1818	VALUE SPACES OCCURS 1000 TIMES.		
1819	05 WS-TBL-TERR-CD	PIC X(05)	
1820	VALUE SPACES OCCURS 1000 TIMES.		
1821	05 WS-TBL-TERR-CD-OLD	PIC X(05)	
1822	VALUE SPACES OCCURS 1000 TIMES.		
1823	05 WS-TBL-LIAB-LT	PIC X(07)	
1824	VALUE SPACES OCCURS 1000 TIMES.		
1825	05 WS-TBL-BI-LT	PIC X(20)	
1826	VALUE SPACES OCCURS 1000 TIMES.		
1827	05 WS-TBL-PD-LT	PIC X(20)	
1828	VALUE SPACES OCCURS 1000 TIMES.		
1829	05 WS-TBL-MED-PAY-LT	PIC X(07)	
1830	VALUE SPACES OCCURS 1000 TIMES.		
1831	05 WS-TBL-MEDEXP-LT	PIC X(07)	
1832	VALUE SPACES OCCURS 1000 TIMES.		
1833	05 WS-TBL-UM-LT	PIC X(07)	
1834	VALUE SPACES OCCURS 1000 TIMES.		
1835	05 WS-TBL-SUM-LT	PIC X(07)	
1836	VALUE SPACES OCCURS 1000 TIMES.		
1837	05 WS-TBL-UI-LT	PIC X(07)	
1838	VALUE SPACES OCCURS 1000 TIMES.		
1839	*		
1840	*-----*		
1841	* THIS TABLE WILL BE USED TO PRINT THE INFORMATION NEEDED ON THE		*
1842	* SCHEDULE OF AUTOMOBILES. THIS IS INFORMATION IS THE VEHICLE PREM.		*
1843	*-----*		
1844	*		
1845	01 WS-SCHEDULE-WRIT-PREM-TABLE.		
1846	05 WS-TBL-LIAB-WRIT	PIC S9(11) COMP-3	
1847	VALUE 0 OCCURS 1000 TIMES.		
1848	05 WS-TBL-PD-WRIT	PIC S9(11) COMP-3	

1849	VALUE 0 OCCURS 1000 TIMES.	
1850	05 WS-TBL-BI-WRIT	PIC S9(11) COMP-3
1851	VALUE 0 OCCURS 1000 TIMES.	
1852	05 WS-TBL-PIP-WRIT	PIC S9(11) COMP-3
1853	VALUE 0 OCCURS 1000 TIMES.	
1854	05 WS-TBL-NY-PIP-WRIT	PIC S9(11) COMP-3
1855	VALUE 0 OCCURS 1000 TIMES.	
1856	05 WS-TBL-ADDED-PIP-WRIT	PIC S9(11) COMP-3
1857	VALUE 0 OCCURS 1000 TIMES.	
1858	05 WS-TBL-PPI-WRIT	PIC S9(11) COMP-3
1859	VALUE 0 OCCURS 1000 TIMES.	
1860	05 WS-TBL-MED-PAY-WRIT	PIC S9(11) COMP-3
1861	VALUE 0 OCCURS 1000 TIMES.	
1862	05 WS-TBL-MEDEXP-WRIT	PIC S9(11) COMP-3
1863	VALUE 0 OCCURS 1000 TIMES.	
1864	05 WS-TBL-MEDEXP	PIC S9(11) COMP-3
1865	VALUE 0 OCCURS 1000 TIMES.	
1866	05 WS-TBL-ILB	PIC S9(11) COMP-3
1867	VALUE 0 OCCURS 1000 TIMES.	
1868	05 WS-TBL-UM-WRIT	PIC S9(11) COMP-3
1869	VALUE 0 OCCURS 1000 TIMES.	
1870	05 WS-TBL-SUM-WRIT	PIC S9(11) COMP-3
1871	VALUE 0 OCCURS 1000 TIMES.	
1872	05 WS-TBL-UI-WRIT	PIC S9(11) COMP-3
1873	VALUE 0 OCCURS 1000 TIMES.	
1874	05 WS-TBL-STATED-AMT	PIC X(06)
1875	VALUE SPACES OCCURS 1000 TIMES.	
1876	05 WS-TBL-COMP-DEDUCT	PIC S9(09) COMP-3
1877	VALUE 0 OCCURS 1000 TIMES.	
1878	05 WS-TBL-COMP-WRIT	PIC S9(11) COMP-3
1879	VALUE 0 OCCURS 1000 TIMES.	
1880	05 WS-TBL-SPEC-LOSS-WRIT	PIC S9(11) COMP-3
1881	VALUE 0 OCCURS 1000 TIMES.	
1882	05 WS-TBL-COLL-DEDUCT	PIC S9(09) COMP-3
1883	VALUE 0 OCCURS 1000 TIMES.	
1884	05 WS-TBL-COLL-WRIT	PIC S9(11) COMP-3
1885	VALUE 0 OCCURS 1000 TIMES.	
1886	05 WS-TBL-WAIVER	PIC X(01)
1887	VALUE SPACES OCCURS 1000 TIMES.	
1888	05 WS-TBL-TOW-WRIT	PIC S9(11) COMP-3
1889	VALUE 0 OCCURS 1000 TIMES.	
1890	05 WS-TBL-TL-PREM	PIC S9(11) COMP-3
1891	VALUE 0 OCCURS 1000 TIMES.	
1892	05 WS-TBL-TL-PREM-WRIT	PIC S9(11) COMP-3
1893	VALUE 0 OCCURS 1000 TIMES.	
1894	05 WS-RENT-PREM	PIC S9(11) COMP-3
1895	VALUE 0 OCCURS 1000 TIMES.	
1896	05 WS-RENT-PREM-WRIT	PIC S9(11) COMP-3
1897	VALUE 0 OCCURS 1000 TIMES.	
1898		
1899	*	
1900	01 WS-SCHEDULE-NET-PREM-TABLE.	
1901	05 WS-TBL-LIAB-NET	PIC S9(11) COMP-3
1902	VALUE 0 OCCURS 1000 TIMES.	
1903	05 WS-TBL-PD-NET	PIC S9(11) COMP-3
1904	VALUE 0 OCCURS 1000 TIMES.	

1905	05 WS-TBL-BI-NET	PIC S9(11) COMP-3
1906	VALUE 0 OCCURS 1000 TIMES.	
1907	05 WS-TBL-PIP-NET	PIC S9(11) COMP-3
1908	VALUE 0 OCCURS 1000 TIMES.	
1909	05 WS-TBL-ADDED-PIP-NET	PIC S9(11) COMP-3
1910	VALUE 0 OCCURS 1000 TIMES.	
1911	05 WS-TBL-OBEL-NET	PIC S9(11) COMP-3
1912	VALUE 0 OCCURS 1000 TIMES.	
1913	05 WS-TBL-PPI-NET	PIC S9(11) COMP-3
1914	VALUE 0 OCCURS 1000 TIMES.	
1915	05 WS-TBL-MED-PAY-NET	PIC S9(11) COMP-3
1916	VALUE 0 OCCURS 1000 TIMES.	
1917	05 WS-TBL-MEDEXP-NET	PIC S9(11) COMP-3
1918	VALUE 0 OCCURS 1000 TIMES.	
1919	05 WS-TBL-UM-NET	PIC S9(11) COMP-3
1920	VALUE 0 OCCURS 1000 TIMES.	
1921	05 WS-TBL-SUM-NET	PIC S9(11) COMP-3
1922	VALUE 0 OCCURS 1000 TIMES.	
1923	05 WS-TBL-UI-NET	PIC S9(11) COMP-3
1924	VALUE 0 OCCURS 1000 TIMES.	
1925	05 WS-TBL-COMP-NET	PIC S9(11) COMP-3
1926	VALUE 0 OCCURS 1000 TIMES.	
1927	05 WS-TBL-SPEC-LOSS-NET	PIC S9(11) COMP-3
1928	VALUE 0 OCCURS 1000 TIMES.	
1929	05 WS-TBL-COLL-NET	PIC S9(11) COMP-3
1930	VALUE 0 OCCURS 1000 TIMES.	
1931	05 WS-TBL-TOW-NET	PIC S9(11) COMP-3
1932	VALUE 0 OCCURS 1000 TIMES.	
1933	05 WS-TBL-ADDED-EXIT	PIC S9(11) COMP-3
1934	VALUE 0 OCCURS 1000 TIMES.	
1935	05 WS-TBL-DROPPED-EXIT	PIC S9(11) COMP-3
1936	VALUE 0 OCCURS 1000 TIMES.	
1937	05 WS-TBL-TL-EXIT	PIC S9(11) COMP-3
1938	VALUE 0 OCCURS 1000 TIMES.	
1939	05 WS-RENT-PREM-NET	PIC S9(11) COMP-3
1940	VALUE 0 OCCURS 1000 TIMES.	
1941	*	
1942	01 WS-EXPOSURE-EXIST-TABLE.	
1943	05 WS-SLL-EXIST	PIC X(01)
1944	VALUE 'N' OCCURS 1000 TIMES.	
1945	05 WS-PIP-EXIST	PIC X(01)
1946	VALUE 'N' OCCURS 1000 TIMES.	
1947	05 WS-NY-PIP-EXIST	PIC X(01)
1948	VALUE 'N' OCCURS 1000 TIMES.	
1949	05 WS-OBEL-EXIST	PIC X(01)
1950	VALUE 'N' OCCURS 1000 TIMES.	
1951	05 WS-ADPIP-EXIST	PIC X(01)
1952	VALUE 'N' OCCURS 1000 TIMES.	
1953	05 WS-MED-EXIST	PIC X(01)
1954	VALUE 'N' OCCURS 1000 TIMES.	
1955	05 WS-MEDEXP-EXIST	PIC X(01)
1956	VALUE 'N' OCCURS 1000 TIMES.	
1957	05 WS-UM-EXIST	PIC X(01)
1958	VALUE 'N' OCCURS 1000 TIMES.	
1959	05 WS-SUM-EXIST	PIC X(01)
1960	VALUE 'N' OCCURS 1000 TIMES.	

1961	05 WS-UI-EXIST	PIC X(01)	
1962	VALUE 'N' OCCURS 1000 TIMES.		
1963	05 WS-COMP-EXIST	PIC X(01)	
1964	VALUE 'N' OCCURS 1000 TIMES.		
1965	05 WS-COLL-EXIST	PIC X(01)	
1966	VALUE 'N' OCCURS 1000 TIMES.		
1967	05 WS-LIMF	PIC X(01)	
1968	VALUE 'N' OCCURS 1000 TIMES.		
1969	05 WS-TOW-EXIST	PIC X(01)	
1970	VALUE 'N' OCCURS 1000 TIMES.		
1971	05 WS-SPEC-LOSS-EXIST	PIC X(01)	
1972	VALUE 'N' OCCURS 1000 TIMES.		
1973	05 WS-BI-EXIST	PIC X(01)	
1974	VALUE 'N' OCCURS 1000 TIMES.		
1975	05 WS-PD-EXIST	PIC X(01)	
1976	VALUE 'N' OCCURS 1000 TIMES.		
1977	05 WS-PPI-EXIST	PIC X(01)	
1978	VALUE 'N' OCCURS 1000 TIMES.		
1979	05 WS-ILB-EXIST	PIC X(01)	
1980	VALUE 'N' OCCURS 1000 TIMES.		
1981	05 WS-LIAB-EXIST	PIC X(01)	
1982	VALUE 'N' OCCURS 1000 TIMES.		
1983	05 WS-BUY-EXIST	PIC X(01)	
1984	VALUE 'N' OCCURS 1000 TIMES.		
1985	05 WS-WAIVE-EXIST	PIC X(01)	
1986	VALUE 'N' OCCURS 1000 TIMES.		
1987	05 WS-RENTL-EXIST	PIC X(01)	
1988	VALUE "N" OCCURS 1000 TIMES.		
1989	*		
1990	*-----*		
1991	* THIS TABLE WILL BE USED TO PRINT THE INFORMATION NEEDED ON THE		*
1992	* SCHEDULE OF LOSS PAYEES.	*	
1993	*-----*		
1994	*		
1995	01 WS-SCH-LOSS-PAYEE-TABLE.		
1996	05 WS-SCH-LP-ACCT	PIC S9(07) COMP-3	VALUE 0
1997	OCCURS 1000 TIMES.		
1998	05 WS-SCH-LP-CAR-NO	PIC 9(03)	VALUE 0
1999	OCCURS 1000 TIMES.		
2000	05 WS-SCH-LP-NAME	PIC X(60)	VALUE SPACES
2001	OCCURS 1000 TIMES.		
2002	05 WS-SCH-LP-ADNAME	PIC X(60)	VALUE SPACES
2003	OCCURS 1000 TIMES.		
2004	05 WS-SCH-LP-STREET1	PIC X(60)	VALUE SPACES
2005	OCCURS 1000 TIMES.		
2006	05 WS-SCH-LP-STREET2	PIC X(60)	VALUE SPACES
2007	OCCURS 1000 TIMES.		
2008	05 WS-SCH-LP-CITY-ST-ZIP	PIC X(36)	VALUE SPACES
2009	OCCURS 1000 TIMES.		
2010	05 WS-SCH-GARAGE-STATE	PIC X(02)	VALUE SPACES
2011	OCCURS 1000 TIMES.		
2012	05 WS-SCH-LP-ORTD	PIC X(03)	VALUE SPACES
2013	OCCURS 1000 TIMES.		
2014	*		
2015	*-----*		
2016	* THIS TABLE WILL BE USED TO PRINT THE INFORMATION NEEDED ON THE		*

```

2017 * NOTICE OF LOSS PAYEES. *
2018 *-----*
2019 *
2020 01 WS-LOSS-PAYEE-TABLE.
2021 05 WS-LP-ACCT PIC S9(07) COMP-3 VALUE 0
2022 OCCURS 1000 TIMES.
2023 05 WS-SV-ENTRY-COUNT PIC 9(03) VALUE 0
2024 OCCURS 1000 TIMES.
2025 05 WS-LP-NAME PIC X(60) VALUE SPACES
2026 OCCURS 1000 TIMES.
2027 05 WS-LP-ADNAME PIC X(60) VALUE SPACES
2028 OCCURS 1000 TIMES.
2029 05 WS-LP-STREET1 PIC X(60) VALUE SPACES
2030 OCCURS 1000 TIMES.
2031 05 WS-LP-STREET2 PIC X(60) VALUE SPACES
2032 OCCURS 1000 TIMES.
2033 05 WS-LP-CITY-ST-ZIP PIC X(36) VALUE SPACES
2034 OCCURS 1000 TIMES.
2035 05 WS-LP-EXP-DATE PIC X(10) VALUE SPACES
2036 OCCURS 1000 TIMES.
2037 05 WS-LP-LIMIT-SPLIT-HOLD PIC X(01) VALUE SPACES
2038 OCCURS 1000 TIMES.
2039 05 WS-LP-LIMIT-HOLD PIC X(20) VALUE SPACES
2040 OCCURS 1000 TIMES.
2041 *
2042 01 WS-APIP-BASIC-LIMIT-NUM PIC 9(07) VALUE 0.
2043 01 WS-PIP-BASIC-LIMIT-NUM PIC 9(07) VALUE 0.
2044 01 WS-OBEL-LIMIT-NUM PIC 9(07) VALUE 0.
2045 *
2046 01 WS-SUM-WRIT-PREM-DEC PIC 9(11)V99 VALUE 0.
2047 01 WS-SUM-WRIT-PREM-ADD-DEC PIC 9(11)V99 VALUE 0.
2048 01 WS-SUM-WRIT-PREM-OBE-DEC PIC 9(11)V99 VALUE 0.
2049 *
2050 *-----*
2051 * SQL WORK FIELD AREA *
2052 *-----*
2053 *
2054 EXEC SQL BEGIN DECLARE SECTION END-EXEC.
2055 *
2056 01 USERID PIC X(01) VALUE "/".
2057 01 DBS PIC X(04) VARYING.
2058 01 WS-DBS-LEN PIC X(01).
2059 01 COMPILED-DATE PIC X(25).
2060 01 WS-USER PIC X(25) VALUE SPACES.
2061 *
2062 *-----*
2063 * SQL COUNTING VARIABLES. *
2064 *-----*
2065 *
2066 01 WS-HOLD-ENTRY-COUNT PIC S9(03) COMP-3 VALUE 0.
2067 01 WS-PREV-ENTRY-COUNT PIC S9(03) COMP-3 VALUE 0.
2068 01 WS-ENTRY-COUNT PIC S9(03) COMP-3 VALUE 0.
2069 01 WS-ENTRY-COUNT-2 PIC S9(03) COMP-3 VALUE 0.
2070 01 WS-ENTRY-COUNT-3 PIC S9(03) COMP-3 VALUE 0.
2071 01 WS-ENTRY-COUNT-4 PIC S9(03) COMP-3 VALUE 0.
2072 01 WS-ENTRY-COUNT-5 PIC S9(03) COMP-3 VALUE 0.

```



```

2073 01 WS-ENTRY-COUNT-6      PIC S9(03) COMP-3 VALUE 0.
2074 01 WS-LP-LINE-COUNT     PIC S9(03) COMP-3 VALUE 0.
2075 01 WS-HIREDAUTO-COUNT    PIC S9(02) COMP-3 VALUE 0.
2076 01 WS-NONOWNED-COUNT    PIC S9(02) COMP-3 VALUE 0.
2077 *
2078 *-----*
2079 *              SWITCHES              *
2080 *-----*
2081 *
2082 01 WS-FIRST-TIME          PIC X(01) VALUE "N".
2083 01 WS-LOSS-PAYEE          PIC X(01) VALUE "N".
2084 01 WS-LOSS-PAYEE-LOCT     PIC X(01) VALUE "N".
2085 *
2086 *-----*
2087 *              CLIENT INFORMATION              *
2088 *-----*
2089 *
2090 01 WS-CLIENT-NAME         PIC X(60) VALUE SPACES.
2091 01 WS-CLIENT-ADNAME       PIC X(60) VALUE SPACES.
2092 01 WS-CLIENT-STREET1      PIC X(60) VALUE SPACES.
2093 01 WS-CLIENT-STREET2      PIC X(60) VALUE SPACES.
2094 01 WS-CLIENT-CITY         PIC X(20) VALUE SPACES.
2095 01 WS-CLIENT-ZIP          PIC X(09) VALUE SPACES.
2096 01 WS-CLIENT-STATE        PIC X(02) VALUE SPACES.
2097 01 WS-CLIENT-BSFM-GB-TYPE PIC X(02) VALUE SPACES.
2098 01 WS-CLIENT-CITY-ST-ZIP  PIC X(36) VALUE SPACES.
2099 01 WS-WORK-CITY           PIC X(23) VALUE SPACES.
2100 01 WS-MI-LOCTS            PIC S9(02) COMP-3 VALUE 0.
2101 *
2102 *-----*
2103 *              VARIABLES FOR INFC TABLE.              *
2104 *-----*
2105 *
2106 01 WS-INFC-GB-EFF-DATE     PIC X(11) VALUE SPACES.
2107 01 WS-INFC-GB-EXP-DATE     PIC X(11) VALUE SPACES.
2108 01 WS-INFC-HSTP-GB-CODE-OF PIC X(02) VALUE SPACES.
2109 01 WS-INFC-GB-ANNUAL-PX    PIC S9(11)V99 COMP-3 VALUE 0.
2110 01 WS-INFC-GB-WRITTEN-PX   PIC S9(11)V99 COMP-3 VALUE 0.
2111 01 WS-INFC-AU-COV-AU-LIAB  PIC X(10) VALUE SPACES.
2112 01 WS-INFC-AU-COV-AU-PIP   PIC X(10) VALUE SPACES.
2113 01 WS-INFC-AU-COV-AU-ADDED-PIP PIC X(10) VALUE SPACES.
2114 01                          PIC X(10) VALUE SPACES.
2115 01 WS-INFC-AU-COV-AU-MED-PAY PIC X(10) VALUE SPACES.
2116 01 WS-INFC-AU-COV-AU-MED-EXP PIC X(10) VALUE SPACES.
2117 01 WS-INFC-AU-COV-AU-UM    PIC X(10) VALUE SPACES.
2118 01 WS-INFC-AU-COV-AU-UI    PIC X(10) VALUE SPACES.
2119 01 WS-INFC-AU-COV-AU-COMP   PIC X(10) VALUE SPACES.
2120 01 WS-INFC-AU-COV-AU-SPECIFIED PIC X(10) VALUE SPACES.
2121 01 WS-INFC-AU-COV-AU-COLLISION PIC X(10) VALUE SPACES.
2122 01 WS-INFC-AU-COV-AU-TOW-LABOR PIC X(10) VALUE SPACES.
2123 01 WS-INFC-AU-COV-AU-BUYBACK PIC X(10) VALUE SPACES.
2124 01 WS-INFC-AU-COV-AU-RENTAL PIC X(10) VALUE SPACES.
2125 01 WS-INFC-AU-COV-AU-PPI    PIC X(10) VALUE SPACES.
2126 01 WS-INFC-GB-END-NO       PIC S9(04) COMP-3 VALUE 0.
2127 01 WS-PREV-INFC-GB-NO      PIC S9(07) COMP-3 VALUE 0.
2128 01 WS-HOLD-PREV-INFC-GB-NO PIC S9(07) COMP-3 VALUE 0.

```

```

2129 01 WS-3000-PREV-INFC-GB-NO PIC S9(07) COMP-3 VALUE 0.
2130 01 WS-INFC-GB-NO PIC S9(07) COMP-3 VALUE 0.
2131 01 WS-INFC-GB-PRIMARY-STATE PIC X(02) VALUE SPACES.
2132 *
2133 *-----*
2134 * VARIABLES FOR BSFM TABLE. *
2135 *-----*
2136 *
2137 01 WS-BSFM-GB-TYPE-DESC PIC X(20) VALUE SPACES.
2138 *
2139 *-----*
2140 * VARIABLES FOR COVERAGE TABLE. *
2141 *-----*
2142 *
2143 01 WS-POLICY-COVERAGE.
2144 05 WS-LIAB-LIMIT PIC X(07) VALUE "0000000"
2145 OCCURS 2 TIMES.
2146 05 WS-MED-PAY-LIMIT PIC X(07) VALUE "0000000"
2147 OCCURS 2 TIMES.
2148 05 WS-MEDEXP-LIMIT PIC X(07) VALUE "0000000"
2149 OCCURS 2 TIMES.
2150 05 WS-UM-LIMIT PIC X(07) VALUE "0000000"
2151 OCCURS 2 TIMES.
2152 05 WS-SUM-LIMIT PIC X(07) VALUE "0000000"
2153 OCCURS 2 TIMES.
2154 05 WS-UI-LIMIT PIC X(07) VALUE "0000000"
2155 OCCURS 2 TIMES.
2156 05 WS-PIP-DEATH-LIMIT PIC X(07) VALUE "0000000"
2157 OCCURS 2 TIMES.
2158 05 WS-PIP-EXP-LIMIT PIC X(07) VALUE "0000000"
2159 OCCURS 2 TIMES.
2160 05 WS-PIP-BASIC-LIMIT PIC X(07) VALUE "0000000"
2161 OCCURS 2 TIMES.
2162 05 WS-NY-PIP-BASIC-LIMIT PIC X(07) VALUE "0000000"
2163 OCCURS 2 TIMES.
2164 05 WS-PIP-WORK-LIMIT PIC X(07) VALUE "0000000"
2165 OCCURS 2 TIMES.
2166 05 WS-OBEL-LIMIT PIC X(07) VALUE "0000000"
2167 OCCURS 2 TIMES.
2168 05 WS-APIP-BASIC-LIMIT PIC X(07) VALUE "0000000"
2169 OCCURS 2 TIMES.
2170 05 WS-NY-APIP-BASIC-LIMIT PIC X(07) VALUE "0000000"
2171 OCCURS 2 TIMES.
2172 05 WS-NY-APDTH-ADD-LIMIT PIC X(07) VALUE "0000000"
2173 OCCURS 2 TIMES.
2174 *
2175 *-----*
2176 * VARIABLES FOR BSTP TABLE. *
2177 *-----*
2178 *
2179 01 WS-BSTP-GB-TYPE-DESC PIC X(40) VALUE SPACES.
2180 *
2181 *-----*
2182 * VARIABLES FOR POLY TABLE. *
2183 *-----*
2184 *

```

```

2185 01 WS-POLY-GB-NO PIC S9(07) COMP-3 VALUE 0.
2186 01 WS-POLY-GB-NO-PART-OF PIC S9(07) COMP-3 VALUE 0.
2187 01 WS-POLY-BSOR-GB-ACCOUNT PIC S9(07) COMP-3 VALUE 0.
2188 01 WS-POLY-BSOR-GB-FM-ACCT PIC S9(07) COMP-3 VALUE 0.
2189 01 WS-POLY-GB-POL-PREFIX PIC X(05) VALUE SPACES.
2190 01 WS-POLY-GB-POL-NUMBER PIC X(06) VALUE SPACES.
2191 01 WS-POLY-GB-EFF-DATE PIC X(11) VALUE SPACES.
2192 01 WS-POLY-EFF-DATE-YYYYMMDD PIC X(8) VALUE SPACES.
2193 01 WS-POLY-GB-EXP-DATE PIC X(11) VALUE SPACES.
2194 01 WS-POLY-GB-CLIENT-NAME PIC X(60) VALUE SPACES.
2195 01 WS-POLY-GB-DBA-NAME PIC X(60) VALUE SPACES.
2196 01 WS-POLY-GB-SYSDATE PIC X(10) VALUE SPACES.
2197 01 WS-POLY-SYSTEM-RATED-IND PIC X(01) VALUE SPACES.
2198 *
2199 01 WS-DAYS-POLICY-PERIOD PIC X(04) VALUE SPACES.
2200 *
2201 *-----*
2202 * VARIABLES FOR LOCT TABLE. *
2203 *-----*
2204 *
2205 01 WS-LOCT-GB-NO PIC S9(07) COMP-3 VALUE 0.
2206 01 WS-LOCT-GB-PB-IND PIC X(01) VALUE SPACES.
2207 01 WS-LOCT-GB-STREET1 PIC X(60) VALUE SPACES.
2208 01 WS-LOCT-GB-STREET2 PIC X(60) VALUE SPACES.
2209 01 WS-LOCT-BSFM-GB-TYPE-CODE-OF PIC X(02) VALUE SPACES.
2210 01 WS-LOCT-ZIPC-GB-NO PIC S9(07) COMP-3 VALUE 0.
2211 01 WS-LOCT-STAT-GB-ALPHA-STATE-IN PIC X(02) VALUE SPACES.
2212 01 WS-LOCT-GB-TERR-CD PIC X(05) VALUE SPACES.
2213 01 WS-COVERAGE-LOCT PIC X(02) VALUE SPACES.
2214 *
2215 01 WS-VINT-BSOR-GB-NO-ENTITY PIC S9(07) COMP-3 VALUE 0.
2216 01 WS-VINT-ORTD-GB-TYPE PIC X(03) VALUE SPACES.
2217 01 WS-VINT-GB-EFF-DATE PIC X(10) VALUE SPACES.
2218 01 WS-VINT-GB-EXP-DATE PIC X(10) VALUE SPACES.
2219 *
2220 01 WS-COVERAGE-STATE PIC X(02) VALUE SPACES.
2221 01 WS-VHCD-AU-NO PIC S9(07) COMP-3 VALUE 0.
2222 01 WS-VHCD-MCTB-AU-CLASS PIC X(02) VALUE SPACES.
2223 *
2224 01 WS-VCLD-STAT-AU-STATE PIC X(02) VALUE SPACES.
2225 01 WS-VCLD-AU-USAGE-TYPE PIC X(02) VALUE SPACES.
2226 01 WS-VCLD-AU-ZONE-RATED PIC X(01) VALUE SPACES.
2227 *
2228 *-----*
2229 * VARIABLES FOR ZIPC TABLE. *
2230 *-----*
2231 *
2232 01 WS-ZIPC-GB-CITY PIC X(20) VALUE SPACES.
2233 01 WS-ZIPC-GB-ZIP PIC X(09) VALUE SPACES.
2234 01 WS-ZIPC-STAT-GB-STATE PIC X(02) VALUE SPACES.
2235 *
2236 01 WS-GARAGE-CITY PIC X(20) VALUE SPACES.
2237 01 WS-GARAGE-STATE PIC X(02) VALUE SPACES.
2238 *
2239 *-----*
2240 * VARIABLES FOR BSOR TABLE. *

```

```

2241 *-----*
2242 *
2243 01 WS-BSOR-GB-ACCOUNT      PIC S9(07) COMP-3 VALUE 0.
2244 01 WS-BSOR-GB-NAME        PIC X(60) VALUE SPACES.
2245 01 WS-BSOR-GB-ADNAME      PIC X(60) VALUE SPACES.
2246 01 WS-BSOR-BSTP-GB-TYPE-CODE-OF PIC X(02) VALUE SPACES.
2247 *
2248 *-----*
2249 *          VARIABLES FOR PPRT TABLE.          *
2250 *-----*
2251 *
2252 01 WS-PPRT-POLY-GB-NO      PIC S9(07) COMP-3 VALUE 0.
2253 01 WS-PPRT-INFC-GB-NO      PIC S9(07) COMP-3 VALUE 0.
2254 *
2255 *-----*
2256 *          VARIABLES FOR AUCV TABLE.          *
2257 *-----*
2258 *
2259 01 WS-AUCV-AU-NO           PIC S9(07) COMP-3 VALUE 0.
2260 01 WS-AUCV-AU-TYPE         PIC X(06) VALUE SPACES.
2261 01 WS-AUCV-AU-SUB-TYPE     PIC X(06) VALUE SPACES.
2262 01 WS-AUCV-AU-LIMIT        PIC X(07) VALUE SPACES.
2263 01 WS-AUCV-AU-LIMIT-REAL   PIC X(20) VALUE SPACES.
2264 01 WS-AUCV-AU-LIMIT-SPLIT  PIC X(01) VALUE SPACES.
2265 01 WS-AUCV-AU-LIMIT-SPLIT-BI PIC X(01) VALUE SPACES.
2266 01 WS-AUCV-AU-LIMIT-SPLIT-PD PIC X(01) VALUE SPACES.
2267 01 WS-AUCV-AU-LIMIT-SPLIT-UM PIC X(01) VALUE SPACES.
2268 01 WS-AUCV-AU-LIMIT-SPLIT-SUM PIC X(01) VALUE SPACES.
2269 01 WS-AUCV-AU-LIMIT-SPLIT-UI PIC X(01) VALUE SPACES.
2270 01 WS-AUCV-AU-SPLIT-LIMIT PIC X(35) VALUE SPACES.
2271 01 WS-AUCV-AU-SPLIT-LIMIT-2 PIC X(35) VALUE SPACES.
2272 01 WS-AUCV-AU-SPLIT-LIMIT-3 PIC X(35) VALUE SPACES.
2273 01 WS-AUCV-AU-LIMIT-SPLIT-HOLD PIC X(01) VALUE SPACES.
2274 01 WS-UM-LIMIT-HOLD        PIC X(20) VALUE SPACES.
2275 01 WS-SUM-LIMIT-HOLD        PIC X(20) VALUE SPACES.
2276 01 WS-UI-LIMIT-HOLD        PIC X(20) VALUE SPACES.
2277 01 WS-BI-LIMIT-HOLD        PIC X(20) VALUE SPACES.
2278 01 WS-PD-LIMIT-HOLD        PIC X(20) VALUE SPACES.
2279 01 WS-AU-LIMIT-SPLIT-HOLD  PIC X(01) VALUE SPACES.
2280 01 WS-AU-LIMIT-HOLD        PIC X(20) VALUE SPACES.
2281 *
2282 *-----*
2283 *          VARIABLES FOR EXIT TABLE.          *
2284 *-----*
2285 *
2286 01 WS-EXIT-GB-NO           PIC S9(11) COMP-3 VALUE 0.
2287 01 WS-EXIT-GB-EFF-DATE     PIC X(11) VALUE SPACES.
2288 01 WS-EXIT-GB-EXP-DATE     PIC X(11) VALUE SPACES.
2289 01 WS-EXIT-VHDS-AU-NO      PIC S9(07) COMP-3 VALUE 0.
2290 01 WS-EXIT-VHCD-AU-NO      PIC S9(07) COMP-3 VALUE 0.
2291 01 WS-EXIT-AUBP-AU-NO      PIC S9(07) COMP-3 VALUE 0.
2292 01 WS-EXIT-NET-PREM        PIC S9(11) COMP-3 VALUE 0.
2293 01 WS-EXIT-WRIT-PREM       PIC S9(11) COMP-3 VALUE 0.
2294 01 WS-EXIT-GB-DEDUCTIBLE   PIC S9(09) COMP-3 VALUE 0.
2295 *
2296 01 WS-SUM-TOTAL-PIP        PIC S9(11)V99 COMP-3 VALUE 0.

```

2297	01	WS-SUM-TL-PIP-LT	PIC S9(07)	COMP-3	VALUE 0.
2298	*				
2299	01	WS-SUM-WRIT-PREM-LIAB	PIC S9(11)	COMP-3	VALUE 0.
2300	01	WS-SUM-WRIT-PREM-BI	PIC S9(11)	COMP-3	VALUE 0.
2301	01	WS-SUM-WRIT-PREM-PD	PIC S9(11)	COMP-3	VALUE 0.
2302	01	WS-SUM-WRIT-PREM-PIP	PIC S9(11)	COMP-3	VALUE 0.
2303	01	WS-NY-WRIT-PREM-PIP	PIC S9(11)	COMP-3	VALUE 0.
2304	01	WS-SUM-WRIT-PREM-OBEL	PIC S9(11)	COMP-3	VALUE 0.
2305	01	WS-SUM-WRIT-PREM-ADDED-PIP	PIC S9(11)	COMP-3	VALUE 0.
2306	01	WS-SUM-WRIT-PREM-NY-ADDED-PIP	PIC S9(11)	COMP-3	VALUE 0.
2307	01	WS-SUM-WRIT-PREM-PPI	PIC S9(11)	COMP-3	VALUE 0.
2308	01	WS-SUM-WRIT-PREM-MED-PAY	PIC S9(11)	COMP-3	VALUE 0.
2309	01	WS-SUM-WRIT-PREM-MEDEXP	PIC S9(11)	COMP-3	VALUE 0.
2310	01	WS-SUM-WRIT-PREM-UM	PIC S9(11)	COMP-3	VALUE 0.
2311	01	WS-SUM-WRIT-PREM-SUM	PIC S9(11)	COMP-3	VALUE 0.
2312	01	WS-SUM-WRIT-PREM-UI	PIC S9(11)	COMP-3	VALUE 0.
2313	01	WS-SUM-WRIT-PREM-COMP	PIC S9(11)	COMP-3	VALUE 0.
2314	01	WS-SUM-WRIT-PREM-COLL	PIC S9(11)	COMP-3	VALUE 0.
2315	01	WS-SUM-WRIT-PREM-SPEC	PIC S9(11)	COMP-3	VALUE 0.
2316	01	WS-SUM-WRIT-PREM-TOW	PIC S9(11)	COMP-3	VALUE 0.
2317	01	WS-SUM-WRIT-PREM-BUYBACK	PIC S9(11)	COMP-3	VALUE 0.
2318	01	WS-SUM-WRIT-PREM-RENTAL	PIC S9(11)	COMP-3	VALUE 0.
2319	01	WS-SUM-NET-PREM-RENTAL	PIC S9(11)	COMP-3	VALUE 0.
2320	01	WS-SUM-WRIT-PREM-HIRED	PIC S9(11)	COMP-3	VALUE 0.
2321	01	WS-SUM-WRIT-PREM-NON-OWNED	PIC S9(11)	COMP-3	VALUE 0.
2322	01	WS-SUM-WRIT-PREM-HIRPHY	PIC S9(11)	COMP-3	VALUE 0.
2323	01	WS-NCRB-WRIT-PREM	PIC S9(11)	COMP-3	VALUE 0.
2324	01	WS-NCRB-LIMIT	PIC S9(07)	COMP-3	VALUE 0.
2325	01	WS-SUM-WRIT-PREM-HIRPHY-1403	PIC X(01)	VALUE	SPACES.
2326	01	WS-SUM-ASSM-MVFEE-1403	PIC X(01)	VALUE	SPACES.
2327	01	WS-PAGE2A-OTHER-PX	PIC S9(09)	COMP-3	VALUE 0.
2328	01	WS-PIP-DEDUCTIBLE	PIC S9(09)	COMP-3	VALUE 0.
2329	01	WS-PIP-DEDUCT	PIC S9(09)	COMP-3	VALUE 0.
2330	01	WS-PPI-DEDUCT	PIC S9(09)	COMP-3	VALUE 0.
2331	*				
2332	*-----*				
2333	*	VARIABLES FOR ASSM TABLE.		*	
2334	*-----*				
2335	*				
2336	01	WS-ASSM-GB-AMOUNT	PIC S9(09)V99	COMP-3	VALUE 0.
2337	01	WS-ASSM-GB-WRIT-AMOUNT	PIC S9(09)V99	COMP-3	VALUE 0.
2338	01	WS-SUM-ASSM-GB-WRIT-AMOUNT	PIC S9(09)V99	COMP-3	VALUE 0.
2339	01	WS-SUM-ASSM-MVFEE	PIC S9(09)V99	COMP-3	VALUE 0.
2340	01	WS-ASSM-CODE-MVFEE	PIC X(10)	VALUE	SPACES.
2341	01	WS-STAS-ASSM-GB-AMOUNT	PIC S9(09)V99	COMP-3	VALUE 0.
2342	01	WS-ASSM-GB-AMOUNT-POLY	PIC S9(09)V99	COMP-3	VALUE 0.
2343	*				
2344	*-----*				
2345	*	VARIABLES FOR ENFM TABLE.		*	
2346	*-----*				
2347	*				
2348	01	WS-ENDFORM	PIC X(20)	VALUE	SPACES.
2349	01	WS-ENDFORM-LEN	PIC S9(03)	COMP-3	VALUE 0.
2350	01	WS-EDITION-DATE	PIC X(05)	VALUE	SPACES.
2351	*				
2352	*-----*				

2353	*	VARIABLES FOR ATCH TABLE.	*
2354	*-----*		
2355	*		
2356	01	WS-ATCH-ENFM-GB-NO	PIC S9(07) COMP-3 VALUE 0.
2357	*-----*		
2358	*		
2359	*	VARIABLES FOR VHDS TABLE.	*
2360	*-----*		
2361	*		
2362	01	WS-VHDS-AU-SERIAL-NO	PIC X(20) VALUE SPACES.
2363	01	WS-VHDS-AU-MAKE	PIC X(20) VALUE SPACES.
2364	01	WS-VHDS-AU-MODEL	PIC X(20) VALUE SPACES.
2365	*-----*		
2366	*		
2367	*	VARIABLES FOR VHCD TABLE.	*
2368	*-----*		
2369	*		
2370	01	WS-VHCD-AU-ID-NO	PIC S9(03) COMP-3 VALUE 0.
2371	01	WS-VHCD-AU-YEAR	PIC S9(04) COMP-3 VALUE 0.
2372	01	WS-VHCD-AU-COST	PIC S9(07) COMP-3 VALUE 0.
2373	01	WS-VHCD-AU-USE	PIC X(02) VALUE SPACES.
2374	01	WS-VHCD-AU-WEIGHT	PIC X(04) VALUE SPACES.
2375	01	WS-VHCD-AU-CLASS-CODE	PIC X(06) VALUE SPACES.
2376	01	WS-VHCD-AU-ZONE-BEGIN	PIC X(05) VALUE SPACES.
2377	*-----*		
2378	*		
2379	*	VARIABLES FOR VHFC TABLE.	*
2380	*-----*		
2381	*		
2382	01	WS-POP-DENSITY	PIC X(04) VALUE SPACES.
2383	01	WS-DRIVER-RECORD	PIC X(12) VALUE SPACES.
2384	01	WS-MILEAGE	PIC X(07) VALUE SPACES.
2385	01	WS-VEHICLE-DROP-EXP	PIC S9(07) COMP-3 VALUE 0.
2386	01	WS-VEHICLE-ADD-EXP	PIC S9(07) COMP-3 VALUE 0.
2387	01	WS-VEHICLE-RV-EXP-COUNT	PIC S9(07) COMP-3 VALUE 0.
2388	01	WS-MCTB-GB-CODE	PIC X(06) VALUE SPACES.
2389	01	WS-MCTB-GB-EDIT	PIC X(16) VALUE SPACES.
2390	*-----*		
2391	*		
2392	*	VARIABLES FOR PLFD TABLE.	*
2393	*-----*		
2394	*		
2395	01	WS-PLFD-GB-MACRO	PIC X(09) VALUE SPACES.
2396	01	WS-PLFD-GB-STATE	PIC X(02) VALUE SPACES.
2397	01	WS-PLFD-GB-FORM-DESC	PIC X(10) VALUE SPACES.
2398	01	WS-PLFD-GB-ENABLE	PIC X(01) VALUE SPACES.
2399	01	WS-PLFD-GB-OPTION-1	PIC X(02) VALUE SPACES.
2400	01	WS-PLFD-GB-OPTION-2	PIC X(02) VALUE SPACES.
2401	01	WS-PLFD-GB-OPTION-OUT	PIC X(02) VALUE SPACES.
2402	01	WS-PLFD-GB-FORM-NO	PIC X(80) VALUE SPACES.
2403	01	WS-HOLD-FORM-NO	PIC X(80) VALUE SPACES.
2404	*-----*		
2405	01	WS-USER1	PIC X(40).
2406	*-----*		
2407	*		
2408	*		

2409	01	WS-ITEM-FOUR-TABLE.		
2410	05	WS-HIRED-AUTO-TEXT	OCCURS 12 TIMES	
2411		VALUE SPACES	PIC X(86).	
2412				
2413	01	WS-ITEM-FIVE-TABLE.		
2414	05	WS-NONOWNED-AUTO-TEXT	OCCURS 12 TIMES	
2415		VALUE SPACES	PIC X(86).	
2416	*			
2417				
2418	01	WS-ASSM-TABLE.		
2419	05	WS-ASSM-GB-ASSMNT-CODE-2A	OCCURS 1000 TIMES	
2420		VALUE SPACES	PIC X(6).	
2421	05	WS-ASDC-GB-DESC-2A	OCCURS 1000 TIMES	
2422		VALUE SPACES	PIC X(30).	
2423	05	WS-ASSM-LOCT-2A	OCCURS 1000 TIMES	
2424		VALUE SPACES	PIC X(02).	
2425	05	WS-ASSM-GB-WRIT-AMOUNT-2A		
2426		OCCURS 1000 TIMES	VALUE 0 PIC S9(7)V99 COMP-3.	
2427	01	WS-KY-ASSM-TABLE.		
2428	05	WS-ASSM-GB-ASSMNT-CODE-KY	OCCURS 1000 TIMES	
2429		VALUE SPACES	PIC X(6).	
2430	05	WS-ASDC-GB-DESC-KY	OCCURS 1000 TIMES	
2431		VALUE SPACES	PIC X(30).	
2432	05	WS-ASSM-LOCT-KY	OCCURS 1000 TIMES	
2433		VALUE SPACES	PIC X(02).	
2434	05	WS-ASSM-GB-WRIT-AMOUNT-KY		
2435		OCCURS 1000 TIMES	VALUE 0 PIC S9(7)V99 COMP-3.	
2436	05	WS-CYCY-GB-CODE-KY		
2437		OCCURS 1000 TIMES	VALUE SPACES PIC X(10).	
2438	*			
2439	01	WS-ASSM-REDUCT-TABLE.		
2440	05	WS-ASSM-REDUCT-DESC	OCCURS 1000 TIMES	
2441		VALUE SPACES	PIC X(60).	
2442	05	WS-ASSM-REDUCT-WRIT-AMOUNT	OCCURS 1000 TIMES	
2443		VALUE 0	PIC S9(7)V99 COMP-3.	
2444	*			
2445	01	WS-ASSM-HOLD-TABLE.		
2446	05	WS-ASDC-GB-DESC	OCCURS 1000 TIMES	
2447		VALUE SPACES	PIC X(60).	
2448	05	WS-ASSM-GB-WRIT-AMT	OCCURS 1000 TIMES	
2449		VALUE 0	PIC S9(7)V99 COMP-3.	
2450	05	WS-CYCY-CDE-KY	OCCURS 1000 TIMES	
2451		VALUE SPACES	PIC X(6).	
2452	*			
2453	01	WS-PRINT-PAGE-2A	PIC S9(7) VALUE 0.	
2454	01	WS-COUNT-ASSM	PIC S9(7) VALUE 0.	
2455	01	WS-ASSM-HOLD	PIC X(6) VALUE SPACES.	
2456	01	WS-HOLD-DESC	PIC X(33) VALUE SPACES.	
2457	01	WS-MCTB-CHECK	PIC X(1) VALUE "N".	
2458	01	WS-CREDIT-COUNT	PIC S9(7) VALUE 0.	
2459	01	WS-ASSM-CODE	PIC X(6) VALUE SPACES.	
2460	01	WS-ASSM-DESC	PIC X(60).	
2461	01	WS-ASSM-DESC-2	REDEFINES WS-ASSM-DESC.	
2462	05	WS-ASSM-STATE	PIC X(02).	
2463	05	FILLER	PIC X(01).	
2464	05	WS-ASSM-DESC-TXT	PIC X(57).	

```

2465 01 WS-ASSM-WRIT          PIC S9(7)V99  COMP-3.
2466 01 WS-CYCY-NAME          PIC X(60)    VALUE SPACES.
2467 01 WS-CYCY-CODE          PIC X(10)    VALUE SPACES.
2468 *
2469 01 WS-UETB-DP-FIELD-NAME   PIC X(60).
2470 01 WS-UETB-DP-PROGRAM-NAME PIC X(30) VALUE "AUDECprt".
2471 01 WS-REPORT-OUT          PIC X(255).
2472
2473 01 WS-PRINT-CSZ            PIC X(3).
2474 01 IDREQ                   PIC X(132).
2475 *-----*
2476 *               END VARIABLES               *
2477 *-----*
2478 *
2479 EXEC SQL DECLARE ORA1 DATABASE END-EXEC.
2480 EXEC SQL END DECLARE SECTION END-EXEC.
2481 *
2482 EXEC SQL
2483     INCLUDE SQLCA5
2484 END-EXEC.
2485 *
2486 PROCEDURE DIVISION.
2487 *
2488 *
2489 0000-MAINLINE.
2490 *
2491     PERFORM 1000-INIT      THRU 1000-EXIT.
2492     PERFORM 2000-PROCESS-CURS THRU 2000-EXIT
2493     UNTIL WS-EOC-DRIVING = 'Y'.
2494     PERFORM 9500-PRT-ERR-MSG THRU 9500-EXIT.
2495     PERFORM 9000-CLOSE      THRU 9000-EXIT.
2496     DISPLAY "ABENDSYM:NO".
2497     STOP RUN.
2498 *
2499 0000-EXIT.
2500     EXIT.
2501 *
2502 *-----*
2503 * THIS PARAGRAPH WILL CONNECT TO ORACLE, SELECT THE USER ID, *
2504 * DECLARE THE DRIVING CURSOR, OPEN THE DRIVING CURSOR, DECLARE *
2505 * THE ATTACH_CURS, DECLARE THE AUTO_CURS, AND FETCH FROM THE *
2506 * DRIVING CURSOR. *
2507 *-----*
2508 *
2509 1000-INIT.
2510 *
2511 ***** LOG PROGRAM ON TO ORACLE.
2512 *
2513     DISPLAY 1 UPON AUG-NUM.
2514     ACCEPT DBS-ARR FROM AUG-VAL.
2515     DISPLAY 2 UPON AUG-NUM.
2516     ACCEPT WS-DBS-LEN FROM AUG-VAL.
2517 *
2518     MOVE WS-DBS-LEN TO DBS-LEN.
2519 *
2520     EXEC SQL

```

```

2521         CONNECT :USERID
2522         AT ORA1 USING :DBS
2523     END-EXEC.
2524 *
2525     MOVE FUNCTION WHEN-COMPILED TO COMPILED-DATE.
2526 *
2527     EXEC SQL AT ORA1
2528         SELECT TO_CHAR(TO_DATE(SUBSTR(:COMPILED-DATE,1,12)
2529 -             , 'YYYYMMDDHH24MI'), 'DD-MON-YYYY HH:MI')
2530         INTO :COMPILED-DATE
2531         FROM DUAL
2532     END-EXEC.
2533 *
2534     DISPLAY COMPILED-DATE.
2535 *
2536     MOVE "OUTFILE" TO WS-UETB-DP-FIELD-NAME.
2537     CALL 'LIBSYS_TRNLOG' USING WS-UETB-DP-FIELD-NAME,
2538                             WS-REPORT-OUT,
2539                             WS-UETB-DP-PROGRAM-NAME.
2540
2541     DISPLAY WS-REPORT-OUT.
2542
2543
2544
2545
2546 *
2547     OPEN OUTPUT REPORT-OUT.
2548 *
2549     EXEC SQL AT ORA1
2550         SELECT USER INTO :WS-USER1 FROM DUAL
2551     END-EXEC.
2552     DISPLAY WS-USER1.
2553 *
2554     IF SQLCODE NOT EQUAL 0
2555         MOVE "*** ERROR ** TRYING TO LOG ON TO ORACLE" TO
2556             WS-ERR-MSG
2557         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
2558     END-IF.
2559 *
2560     EXEC SQL AT ORA1 EXECUTE
2561         BEGIN
2562             P_ALTER-NLS_YYYY;
2563         END;
2564     END-EXEC.
2565 *
2566     IF SQLCODE NOT = 0 THEN
2567         MOVE "*** ERROR ** IN P_ALTER-NLS_YYYY" TO WS-ERR-MSG
2568         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
2569     END-IF.
2570 *
2571     EXEC SQL AT ORA1
2572         SELECT SUBSTR(USER,5,30)
2573         INTO :WS-USER
2574         FROM DUAL
2575     END-EXEC.
2576 *

```

```

2577 IF SQLCODE NOT EQUAL 0
2578     MOVE "*** ERROR ** SELECTING USER ID " TO
2579         WS-ERR-MSG
2580     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
2581 END-IF.
2582 *
2583 EXEC SQL AT ORA1 DECLARE CURS_PRINT CURSOR FOR
2584     SELECT PPRT_POLY_GB_NO,
2585         PPRT_INFC_GB_NO
2586     FROM PPRT
2587     WHERE PPRT_GB_USER = :WS-USER
2588     AND PPRT_GB_DONE = 'D'
2589     AND PPRT_GB_POL_TYPE = 'A'
2590 END-EXEC.
2591 *
2592 EXEC SQL
2593     OPEN CURS_PRINT
2594 END-EXEC.
2595 *
2596 IF SQLCODE NOT EQUAL 0
2597     MOVE "*** ERROR ** DECLARING CURS_PRINT CURSOR " TO
2598         WS-ERR-MSG
2599     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
2600 END-IF.
2601 *
2602 PERFORM 4000-FETCH-CURS THRU 4000-EXIT.
2603 *
2604 EXEC SQL AT ORA1 DECLARE ATTACH_CURS CURSOR FOR
2605     SELECT ATCH_ENFM_GB_NO
2606     FROM ATCH, ENFM
2607     WHERE ATCH_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2608     AND ATCH_ENFM_GB_NO = ENFM_GB_NO
2609     AND ENFM_GB_MAND_OPT IN('M','O')
2610     ORDER BY ATCH_ENFM_GB_FORM_NO
2611 END-EXEC.
2612 *
2613 EXEC SQL AT ORA1 DECLARE AUTO_CURS CURSOR FOR
2614     SELECT DISTINCT(EXIT_VHCD_AU_NO),
2615         EXIT_VHDS_AU_NO,
2616         LOCT_ZIPC_GB_NO,
2617         LOCT_STAT_GB_ALPHA_STATE_IN,
2618         LOCT_GB_NO,
2619         LOCT_GB_TERR_CD,
2620         VHCD_AU_ID_NO
2621     FROM LOCT, VHCD, EXIT, COIT
2622     WHERE COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2623     AND EXIT_GB_NO = COIT_EXIT_GB_NO
2624     AND VHCD_AU_NO = EXIT_VHCD_AU_NO
2625     AND LOCT_GB_NO = COIT_LOCT_GB_NO
2626     AND EXIT_VHDS_AU_NO IS NOT NULL
2627     AND EXIT_GB_NO NOT IN
2628         (SELECT COIT_EXIT_GB_NO
2629         FROM COIT
2630         WHERE COIT_INFC_GB_NO = :WS-PREV-INFC-GB-NO)
2631     ORDER BY LOCT_STAT_GB_ALPHA_STATE_IN,
2632         VHCD_AU_ID_NO

```

```

2633      END-EXEC.
2634  *
2635  EXEC SQL AT ORA1
2636      DECLARE VINT_CURS CURSOR FOR
2637      SELECT VINT_BSOR_GB_NO_ENTITY,
2638             VINT_ORTD_GB_TYPE,
2639             TO_CHAR(VINT_GB_EFF_DATE,'MM/DD/YYYY'),
2640             TO_CHAR(VINT_GB_EXP_DATE,'MM/DD/YYYY')
2641      FROM VINT
2642      WHERE VINT_BSOR_GB_NO_CLIENT =
2643             :WS-POLY-BSOR-GB-ACCOUNT
2644      AND VINT_VHCD_AU_NO      = :WS-VHCD-AU-NO
2645      AND ((VINT_GB_EFF_DATE  != VINT_GB_EXP_DATE
2646      AND :WS-INFC-HSTP-GB-CODE-OF IN
2647             ('NB', 'FR', 'RB', 'RW'))
2648      OR :WS-INFC-HSTP-GB-CODE-OF NOT IN
2649             ('NB', 'FR', 'RB', 'RW'))
2650      AND VINT_GB_EFF_DATE
2651      <= TO_DATE(:WS-INFC-GB-EFF-DATE,'MM/DD/YYYY')
2652      AND TO_DATE(NVL(VINT_GB_EXP_DATE,'31-DEC-2100'))
2653      >= TO_DATE(:WS-INFC-GB-EFF-DATE,'MM/DD/YYYY')
2654      AND VINT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2655      AND VINT_ORTD_GB_TYPE != 'AIO'
2656  END-EXEC.
2657  *
2658  EXEC SQL AT ORA1
2659      DECLARE AUTO_PREM_CURS CURSOR FOR
2660      SELECT EXIT_GB_NO,
2661             EXIT_GB_NET_PREM,
2662             EXIT_GB_WRIT_PREM,
2663             EXIT_GB_DEDUCTIBLE,
2664             DECODE(AUCV_AU_TYPE,
2665                    'MCSLL','SLL',
2666                    'ASLL', 'SLL',
2667                    'ASPL', 'SLL',
2668                    'IPIP', 'PIP',
2669                    'APIP', 'PIP',
2670                    'PIPST','PIP',
2671                    'PIPI', 'PIP',
2672                    'GPIP','PIP',
2673                    'PEDPIP','PIP',
2674                    'EXPIP', 'PIP',
2675                    'AAPIP', 'ADPIP',
2676                    'ADPIP',DECODE(LOCT_STAT_GB_ALPHA_STATE_IN,
2677                    'NY','NYPIP','ADPIP'),
2678                    'AUM', 'UM',
2679                    'INIUM', 'UM',
2680                    'INNUM', 'UM',
2681                    'INSUM', 'UM',
2682                    'NFUM', 'UM',
2683                    'NNFUM', 'UM',
2684                    'UMS', 'UM',
2685                    'UMN', 'UM',
2686                    'UMUI', 'UM',
2687                    'UMBI', 'UM',
2688                    'UMIN', 'UM',

```

2689	'UMOT', 'UM',
2690	'UMINPD','UM',
2691	'UMOTPD','UM',
2692	'UMUIIN','UM',
2693	'UMUIOT','UM',
2694	'UMBIPD','UM',
2695	'UMINAO','UM',
2696	'UMOTAO','UM',
2697	'UMINRB','UM',
2698	'UMOTRB','UM',
2699	'UM',DECODE(LOCT_STAT_GB_ALPHA_STATE_IN,
2700	'NY','SUM','UM'),
2701	'AUMPD', 'UMPD',
2702	'UIPD', 'UMPD',
2703	'AUI', 'UI',
2704	'UICC', 'UI',
2705	'UIS', 'UI',
2706	'UIN', 'UI',
2707	'UIOT', 'UI',
2708	'UIIN', 'UI',
2709	'UIPDOT','UI',
2710	'UIPDIN','UI',
2711	'UICCN','UI',
2712	'UICCOT','UI',
2713	'UIBIPD','UI',
2714	'AMED', 'MED',
2715	'ILB', DECODE(:WS-CLIENT-STATE,'VA',
2716	'ILB','MED'),
2717	'ACOLL', 'COLL',
2718	'COLWG', 'COLL',
2719	'ACOMP', 'COMP',
2720	'CMPWG', 'COMP',
2721	'GCOM','GLASSM',
2722	'GCOL', DECODE (:WS-CLIENT-STATE,'VA',
2723	'COLL','GLASSL'),
2724	'A6601', '6601',
2725	'CPIP', 'PIP',
2726	'EXMED', 'PIP',
2727	'UMSIN','UM',
2728	'UMSOT', 'UM',
2729	'UMNIN', 'UM',
2730	'UMNOT', 'UM',
2731	'UISIN', 'UI',
2732	'UISOT', 'UI',
2733	'UININ', 'UI',
2734	'UINOT', 'UI',
2735	'NOUM', 'UM',
2736	'NOUI', 'UI',
2737	'HUM', 'UM',
2738	'HUI', 'UI',
2739	'PIPF', 'PIP',
2740	'PIPO', 'PIP',
2741	'PIPOB', 'PIP',
2742	'OBWRK', 'PIP',
2743	'OBAE', 'PIP',
2744	'OBFE', 'PIP',


```

2745         'OBDB', 'PIP',
2746         'SPL', 'BI',
2747         'SPLPD', 'PD',
2748         'UUBN', 'UM',
2749         'UUBPN', 'UM',
2750         'UUBS', 'UM',
2751         'UUBPS', 'UM',
2752         'BROD', 'BROAD',
2753         'LIMO', 'LIMIT',
2754         'TXSDP', 'DSPIP',
2755         'TXSDM', 'DSMED',
2756         AUCV_AU_TYPE, AUCV_AU_TYPE),
2757         AUCV_AU_SUB_TYPE,
2758         SUBSTR(DECODE(AUCV_AU_LIMIT,
2759         'ACV', 'ACV',
2760         'RC', 'RC',
2761         'STATED', 'STATED',
2762         LPAD(AUCV_AU_LIMIT, 7, '0')), 1, 7),
2763         LPAD(AUCV_AU_LIMIT, 7, '0'),
2764         AUCV_AU_LIMIT_SPLIT,
2765         NVL(EXIT_AUBP_AU_NO, 0),
2766         TO_CHAR(EXIT_GB_EFF_DATE, 'MM/DD/YYYY'),
2767         TO_CHAR(EXIT_GB_EXP_DATE, 'MM/DD/YYYY'),
2768         AUCV_AU_LIMIT,
2769         LOCT_STAT_GB_ALPHA_STATE_IN
2770 FROM AUCV, EXIT, COIT, LOCT
2771 WHERE AUCV_AU_NO = EXIT_AUCV_AU_NO
2772 AND EXIT_VHDS_AU_NO = :WS-EXIT-VHDS-AU-NO
2773 AND EXIT_VHCD_AU_NO = :WS-EXIT-VHCD-AU-NO
2774 AND EXIT_GB_NO = COIT_EXIT_GB_NO
2775 AND COIT_LOCT_GB_NO = LOCT_GB_NO
2776 AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2777 AND EXIT_GB_NO NOT IN
2778     (SELECT COIT_EXIT_GB_NO
2779     FROM COIT
2780     WHERE COIT_INFC_GB_NO
2781     = :WS-PREV-INFC-GB-NO)
2782 ORDER BY EXIT_GB_NET_PREM
2783 END-EXEC.
2784
2785 *
2786 EXEC SQL AT ORA1
2787 DECLARE OTHER_COV_CURS CURSOR FOR
2788     SELECT EXIT_WC_CLASS_CODE,
2789           SUBSTR(MCTB_GB_DESC, 1, 60),
2790           NVL(EXIT_GB_NET_PREM, 0)
2791 FROM MCTB, EXIT
2792 WHERE EXIT_GB_NO IN
2793     (SELECT COIT_EXIT_GB_NO
2794     FROM COIT
2795     WHERE COIT_INFC_GB_NO
2796     = :WS-PPRT-INFC-GB-NO)
2797 AND MCTB_GB_CODE_TYPE = 'AUCOV'
2798 AND MCTB_GB_EDIT = EXIT_WC_CLASS_CODE
2799 AND EXIT_WC_CLASS_CODE
2800     IN
2801     ('LERO', 'FENAM', 'FEDES', 'SUBRO', 'GKCOLL', 'GKCOMP')

```

```

2801 UNION
2802 SELECT 'INTER', --THIS IS NOT USED OR PRINTED ANYWHERE
2803         SUBSTR(MCTB_GB_DESC,1,19),
2804         SUM(NVL(EXIT_GB_NET_PREM,0))
2805 FROM MCTB,EXIT
2806 WHERE EXIT_GB_NO IN
2807         (SELECT COIT_EXIT_GB_NO
2808          FROM COIT
2809          WHERE COIT_INFC_GB_NO
2810               = :WS-PPRT-INFC-GB-NO)
2811 AND MCTB_GB_CODE_TYPE = 'AUCOV'
2812 AND MCTB_GB_EDIT = EXIT_WC_CLASS_CODE
2813 AND SUBSTR(EXIT_WC_CLASS_CODE,1,4)
2814      IN ('TICL','TICM','TISP')
2815 GROUP BY SUBSTR(MCTB_GB_DESC,1,19)
2816 END-EXEC.
2817 *
2818 EXEC SQL AT ORA1
2819 DECLARE LOC_STAT_CURS CURSOR FOR
2820 SELECT DISTINCT LOCT_STAT_GB_ALPHA_STATE_IN
2821 FROM LOC, COIT, INFC,LOPM
2822 WHERE LOCT_GB_NO = COIT_LOCT_GB_NO
2823 AND COIT_INFC_GB_NO = INFC_GB_NO
2824 AND INFC_GB_NO = LOPM_INFC_GB_NO
2825 AND INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2826 AND LOPM_GB_WRIT_PX > 0
2827 END-EXEC.
2828
2829 EXEC SQL AT ORA1
2830 DECLARE UMUI_CURS CURSOR FOR
2831 SELECT LOCT_STAT_GB_ALPHA_STATE_IN AS CVGSTATE,
2832        COUNT (DISTINCT DECODE (AUCV_AU_TYPE, 'UI',
2833                                'UNINSURED', 'UM', 'UNDERINSURED')) AS CVGDESC
2834 FROM AUTO_COVERAGE AUCV,
2835      EXPOSURE EXIT ,
2836      COVERED_ITEM COIT ,
2837      LOC LOCT
2838 WHERE AUCV_AU_NO = EXIT_AUCV_AU_NO
2839 AND COIT_EXIT_GB_NO = EXIT_GB_NO
2840 AND COIT_LOCT_GB_NO = LOCT_GB_NO
2841 AND (AUCV_AU_TYPE LIKE ('%UM%')
2842      OR AUCV_AU_TYPE LIKE ('%UI%'))
2843 AND EXIT_GB_NO IN
2844      (SELECT COIT_EXIT_GB_NO
2845       FROM COIT
2846        WHERE COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2847      )
2848 AND LOCT.LOCT_STAT_GB_ALPHA_STATE_IN = :WS-LOC-STATE
2849 GROUP BY LOCT.LOCT_STAT_GB_ALPHA_STATE_IN
2850 END-EXEC.
2851
2852
2853
2854 EXEC SQL AT ORA1
2855 DECLARE DETERMINE_UMUI CURSOR FOR
2856 SELECT DISTINCT SUBSTR (AUCV_AU_TYPE, 1, 2) AS ACODE,

```

```

2857             SUBSTR (AUCV_AU_TYPE, 3, 2) AS ATYPE
2858 FROM AUTO_COVERAGE AUCV,
2859     EXPOSURE EXIT ,
2860     COVERED_ITEM COIT ,
2861     LOC LOCT
2862 WHERE AUCV_AU_NO   = EXIT_AUCV_AU_NO
2863 AND COIT_EXIT_GB_NO = EXIT_GB_NO
2864 AND COIT_LOCT_GB_NO = LOCT_GB_NO
2865 AND (AUCV_AU_TYPE LIKE ('%UM%')
2866      OR AUCV_AU_TYPE LIKE ('%UI%'))
2867 AND EXIT_GB_NO   IN
2868     (SELECT COIT_EXIT_GB_NO
2869      FROM COIT
2870       WHERE COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2871      )
2872 AND LOCT.LOCT_STAT_GB_ALPHA_STATE_IN = :WS-LOC-STATE
2873 END-EXEC.
2874
2875 *
2876 EXEC SQL AT ORA1
2877 DECLARE DETERMINE_TXD CURSOR FOR
2878 SELECT
2879     AUCV_AU_TYPE ,
2880     SUM(EXIT_GB_NET_PREM),
2881     AUCV_AU_LIMIT
2882 FROM
2883     AUTO_COVERAGE AUCV,
2884     EXPOSURE EXIT ,
2885     COVERED_ITEM COIT
2886 WHERE
2887     AUCV_AU_NO   = EXIT_AUCV_AU_NO
2888 AND COIT_EXIT_GB_NO = EXIT_GB_NO
2889 AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
2890 AND AUCV_AU_TYPE LIKE ('TXSD%')
2891 GROUP BY
2892     AUCV_AU_TYPE ,
2893     AUCV_AU_LIMIT ,
2894     EXIT_GB_STD_PREM
2895 END-EXEC.
2896
2897
2898 *
2899 MOVE 1 TO WS-ENTRY-COUNT-3.
2900 MOVE 1 TO WS-ENTRY-COUNT-4.
2901 *
2902 MOVE SPACES TO PRINT-LINE.
2903 WRITE PRINT-LINE FROM IDREQFNT
2904 AFTER ADVANCING PAGE.
2905 STRING "Print Requested By ", WS-USER1
2906 DELIMITED BY SIZE INTO IDREQ.
2907 WRITE PRINT-LINE FROM IDREQ
2908 AFTER ADVANCING 1.
2909 *
2910 1000-EXIT.
2911 EXIT.
2912 *
```

```

2913 *-----*
2914 * THIS PARAGRAPH WILL BE PERFORMED UNTIL THE END OF THE DRIVING *
2915 * CURSOR IS REACHED. THIS PARAGRAPH SELECTS POLICY INFO, PRINTS *
2916 * THE AUTO POLICY, DELETES THE RECORD JUST PROCESSED FROM THE *
2917 * PPRT TABLE AND WILL FETCH THE RECORD FROM THE DRIVING CURSOR. *
2918 *-----*
2919 *
2920 2000-PROCESS-CURS.
2921
2922 *
2923     PERFORM 5000-REINIT-VAR      THRU 5000-EXIT.
2924 *
2925     PERFORM 2100-GET-POLICY-INFO THRU 2100-EXIT.
2926 *
2927     EVALUATE WS-INFC-HSTP-GB-CODE-OF
2928         WHEN 'NP'
2929             MOVE 1 TO WS-ENTRY-COUNT-3
2930             PERFORM 7600-PRINT-LP-NOTICE THRU 7600-EXIT
2931                 UNTIL WS-LP-ACCT(WS-ENTRY-COUNT-3) = 0
2932         WHEN 'RV'
2933             COMPUTE WS-ENTRY-COUNT = WS-ENTRY-COUNT - 1
2934             MOVE WS-ENTRY-COUNT TO WS-PRT-RV-NO-OF
2935             MOVE 1 TO WS-ENTRY-COUNT
2936             PERFORM 7800-PRINT-RV-POLICY THRU 7800-EXIT
2937                 UNTIL WS-TBL-VHDS-AU-NO(WS-ENTRY-COUNT) = 0
2938             MOVE 1 TO WS-ENTRY-COUNT-3
2939             PERFORM 7600-PRINT-LP-NOTICE THRU 7600-EXIT
2940                 UNTIL WS-LP-ACCT(WS-ENTRY-COUNT-3) = 0
2941         WHEN OTHER
2942             PERFORM 7000-PRINT-AUTO-POLICY THRU 7000-EXIT
2943     END-EVALUATE.
2944 *
2945 *
2946     MOVE SPACES TO WS-AUCV-AU-LIMIT-SPLIT,
2947                 WS-AUCV-AU-LIMIT-SPLIT-BI,
2948                 WS-AUCV-AU-LIMIT-SPLIT-PD,
2949                 WS-AUCV-AU-LIMIT-SPLIT-UM,
2950                 WS-AUCV-AU-LIMIT-SPLIT-SUM,
2951                 WS-AUCV-AU-LIMIT-SPLIT-UI,
2952     WS-AUCV-AU-LIMIT-SPLIT-HOLD,
2953                 WS-BI-LIMIT-HOLD,
2954                 WS-PD-LIMIT-HOLD,
2955                 WS-UM-LIMIT-HOLD,
2956                 WS-SUM-LIMIT-HOLD,
2957                 WS-UI-LIMIT-HOLD.
2958
2959     MOVE 'N' TO WS-NY-EXIST,
2960                 WS-PA-EXIST,
2961                 WS-VA-EXIST,
2962                 WS-MI-EXIST,
2963                 WS-NC-EXIST,
2964                 WS-KY-EXIST,
2965                 WS-WA-EXIST,
2966                 WS-WA-UIPD-EXIST,
2967                 WS-STUM-EXIST,
2968                 WS-NY-SUM-EXIST,

```

```
2969
2970 *
2971     PERFORM 4000-FETCH-CURS   THRU 4000-EXIT.
2972 *
2973     2000-EXIT.
2974     EXIT.
2975 *
2976 * -----*
2977 *   THIS PARAGRAPH WILL PERFORM PARAGRAPHS WHICH WILL SELECT THE  *
2978 *   INFORMATION NEEDED TO PRINT ON AN AUTO POLICY.                *
2979 * -----*
2980 *
2981     2100-GET-POLICY-INFO.
2982
2983 *
2984     PERFORM 2200-GET-DEC-PAGE-INFO   THRU 2200-EXIT.
2985 *
2986     PERFORM 2250-GET-LOCATION-INFO   THRU 2250-EXIT.
2987 *
2988     PERFORM 2300-GET-COVERAGE-INFO   THRU 2300-EXIT
2989 *
2990     PERFORM 2400-GET-ADD-POLICY-INFO THRU 2400-EXIT.
2991 *
2992     2100-EXIT.
2993     EXIT.
2994 *
2995 * -----*
2996 *   THIS PARAGRAPH WILL SELECT INFORMATION FROM THE POLY, BSTP,  *
2997 *   AND BSFM TABLES NEEDED TO PRINT ON THE AUTO POLICY DEC PAGE. *
2998 * -----*
2999 *
3000     2200-GET-DEC-PAGE-INFO.
3001
3002
3003 *
3004     INITIALIZE
3005         WS-POLY-BSOR-GB-ACCOUNT
3006         WS-POLY-GB-EFF-DATE,
3007         WS-POLY-GB-EXP-DATE,
3008         WS-POLY-GB-NO,
3009         WS-POLY-GB-POL-NUMBER,
3010         WS-POLY-GB-POL-PREFIX,
3011         WS-POLY-GB-CLIENT-NAME,
3012         WS-POLY-GB-DBA-NAME,
3013         WS-POLY-GB-SYSDATE,
3014         WS-DAYS-POLICY-PERIOD,
3015         WS-POLY-SYSTEM-RATED-IND.
3016 *
3017     EXEC SQL AT ORA1
3018         SELECT POLY_BSOR_GB_ACCOUNT_OWNED_BY,
3019             POLY_BSOR_GB_FM_ACCT,
3020             TO_CHAR(POLY_GB_EFF_DATE,'MM/DD/YYYY'),
3021             TO_CHAR(POLY_GB_EXP_DATE,'MM/DD/YYYY'),
3022             POLY_GB_NO,
3023             POLY_GB_POL_NUMBER,
3024             POLY_GB_POL_PREFIX,
```

```

3025         UPPER(POLY_GB_CLIENT_NAME),
3026         UPPER(POLY_GB_DBA_NAME),
3027         TO_CHAR(SYSDATE,'MM/DD/YYYY'),
3028         TO_CHAR(POLY_GB_EXP_DATE - POLY_GB_EFF_DATE),
3029         NVL(DECODE(SYSTEM_RATED_IND,'H','H','C'),'C')
3030     INTO :WS-POLY-BSOR-GB-ACCOUNT,
3031         :WS-POLY-BSOR-GB-FM-ACCT,
3032         :WS-POLY-GB-EFF-DATE,
3033         :WS-POLY-GB-EXP-DATE,
3034         :WS-POLY-GB-NO,
3035         :WS-POLY-GB-POL-NUMBER,
3036         :WS-POLY-GB-POL-PREFIX,
3037         :WS-POLY-GB-CLIENT-NAME,
3038         :WS-POLY-GB-DBA-NAME,
3039         :WS-POLY-GB-SYSDATE,
3040         :WS-DAYS-POLICY-PERIOD,
3041         :WS-POLY-SYSTEM-RATED-IND
3042     FROM POLY
3043     WHERE :WS-PPRT-POLY-GB-NO = POLY_GB_NO
3044     END-EXEC.
3045 *
3046     IF SQLCODE NOT EQUAL 0
3047         MOVE "*** ERROR ** SELECTING POLY INFO - 2200"
3048         TO WS-ERR-MSG
3049         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3050         GO TO 2200-EXIT
3051     END-IF.
3052 *
3053     EXEC SQL AT ORA1
3054         SELECT TO_CHAR(TO_DATE(:WS-POLY-GB-EFF-DATE
3055 -             , 'MM/DD/YYYY'), 'YYYYMMDD')
3056         INTO :WS-POLY-EFF-DATE-YYYYMMDD
3057         FROM DUAL
3058     END-EXEC.
3059 *
3060     MOVE SPACES TO WS-POLY-NO.
3061     STRING WS-POLY-GB-POL-PREFIX
3062         DELIMITED BY " ",
3063         "- ",
3064         WS-POLY-GB-POL-NUMBER
3065         DELIMITED BY SIZE
3066     INTO WS-POLY-NO.
3067 *
3068     MOVE WS-POLY-BSOR-GB-ACCOUNT TO WS-BSOR-GB-ACCOUNT.
3069     MOVE 'X' TO WS-LOCT-GB-PB-IND.
3070     MOVE 'N' TO WS-LOSS-PAYEE-LOCT.
3071 *
3072     PERFORM 2800-GET-ACCT-INFO THRU 2800-EXIT.
3073 *
3074     MOVE WS-BSOR-GB-NAME      TO WS-CLIENT-NAME.
3075     MOVE WS-BSOR-GB-ADNAME    TO WS-CLIENT-ADNAME.
3076     MOVE WS-LOCT-GB-STREET1   TO WS-CLIENT-STREET1.
3077     MOVE WS-LOCT-GB-STREET2   TO WS-CLIENT-STREET2.
3078     MOVE WS-LOCT-BSFM-GB-TYPE-CODE-OF
3079     TO WS-CLIENT-BSFM-GB-TYPE.
3080     MOVE WS-ZIPC-GB-CITY      TO WS-CLIENT-CITY.

```



```

3081      MOVE WS-ZIPC-GB-ZIP      TO WS-CLIENT-ZIP.
3082      MOVE WS-ZIPC-STAT-GB-STATE TO WS-CLIENT-STATE.
3083      MOVE SPACES              TO WS-CLIENT-CITY-ST-ZIP.
3084  *
3085      MOVE WS-CLIENT-CITY  TO WS-WORK-CITY.
3086      IF WS-CLIENT-ZIP(6:4) = SPACES
3087          STRING WS-WORK-CITY DELIMITED BY " "
3088              " ",
3089              WS-CLIENT-STATE,
3090              " ",
3091              WS-CLIENT-ZIP(1:5)
3092          DELIMITED BY SIZE
3093      INTO WS-CLIENT-CITY-ST-ZIP
3094  ELSE
3095      STRING WS-WORK-CITY DELIMITED BY " "
3096          " ",
3097          WS-CLIENT-STATE,
3098          " ",
3099          WS-CLIENT-ZIP(1:5),
3100          "-",
3101          WS-CLIENT-ZIP(6:4)
3102      DELIMITED BY SIZE
3103      INTO WS-CLIENT-CITY-ST-ZIP
3104  END-IF.
3105  *
3106      IF WS-POLY-GB-CLIENT-NAME(1:1) = '*'
3107          MOVE SPACES TO WS-CLIENT-NAME
3108      END-IF.
3109  *
3110      IF WS-POLY-GB-CLIENT-NAME(1:1) NOT EQUAL '*'
3111          AND WS-POLY-GB-CLIENT-NAME NOT EQUAL SPACES
3112          MOVE WS-POLY-GB-CLIENT-NAME TO WS-CLIENT-NAME
3113      END-IF.
3114  *
3115      IF WS-POLY-GB-DBA-NAME(1:1) = '*'
3116          MOVE SPACES TO WS-CLIENT-ADNAME
3117      END-IF.
3118  *
3119      IF WS-POLY-GB-DBA-NAME(1:1) NOT EQUAL '*'
3120          AND WS-POLY-GB-DBA-NAME NOT EQUAL SPACES
3121          MOVE WS-POLY-GB-DBA-NAME TO WS-CLIENT-ADNAME
3122      END-IF.
3123  *
3124      INITIALIZE
3125          WS-BSFM-GB-TYPE-DESC.
3126  *
3127      EXEC SQL AT ORA1
3128          SELECT INITCAP(LOWER(BSFM_GB_TYPE_DESC))
3129          INTO :WS-BSFM-GB-TYPE-DESC
3130          FROM BSFM
3131          WHERE BSFM_GB_TYPE_CODE = :WS-CLIENT-BSFM-GB-TYPE
3132      END-EXEC.
3133  *
3134      IF SQLCODE NOT EQUAL 0
3135          MOVE "*** ERROR ** SELECTING BSFM DESC - 2200" TO
3136              WS-ERR-MSG

```

```

3137         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3138         GO TO 2200-EXIT
3139     END-IF.
3140 *
3141     INITIALIZE
3142         WS-BSTP-GB-TYPE-DESC.
3143 *
3144     EXEC SQL AT ORA1
3145         SELECT REPLACE(Initcap(BSTP_GB_TYPE_DESC), 'Bypass', null)
3146         INTO :WS-BSTP-GB-TYPE-DESC
3147         FROM BSTP
3148         WHERE BSTP_GB_TYPE_CODE = :WS-BSOR-BSTP-GB-TYPE-CODE-OF
3149     END-EXEC.
3150 *
3151     IF SQLCODE NOT EQUAL 0
3152         MOVE "*** ERROR ** SELECTING BSTP DESC - 2200" TO
3153         WS-ERR-MSG
3154         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3155     END-IF.
3156 *
3157 *
3158     2200-EXIT.
3159     EXIT.
3160 *
3161 * -----*
3162 * THIS PARAGRAPH DETERMINES THE LOCATION STATES WITH PREMIUM *
3163 * TO HELP DETERMINE WHAT INFORMATION SHOULD PRINT ON THE DEC *
3164 * -----*
3165 *
3166     2250-GET-LOCATION-INFO.
3167 *
3168     EXEC SQL AT ORA1
3169         OPEN LOC_STAT_CURS
3170     END-EXEC.
3171
3172     IF SQLCODE NOT EQUAL 0
3173         AND SQLCODE NOT EQUAL 1403
3174         MOVE "*** ERROR ** OPENING LOC_STAT_CURS CUROS"
3175         TO WS-ERR-MSG
3176         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3177     END-IF.
3178 *
3179     PERFORM 2260-FETCH-LOC-STATE THRU 2260-EXIT
3180     UNTIL WS-EOC-LOC-STATE = "Y".
3181 *
3182     MOVE "N" TO WS-EOC-LOC-STATE.
3183 *
3184     IF WS-STATE-COUNT > 1 THEN
3185         MOVE "Y" TO WS-MULTISTATE
3186     ELSE
3187         MOVE "N" TO WS-MULTISTATE
3188     END-IF.
3189 *
3190     2250-EXIT.
3191 *
3192     2260-FETCH-LOC-STATE.

```

```

3193      *
3194      INITIALIZE
3195      WS-LOC-STATE.
3196      *
3197      EXEC SQL
3198      FETCH LOC_STAT_CURS
3199      INTO :WS-LOC-STATE
3200      END-EXEC.
3201
3202
3203      MOVE ZERO TO EOF-UMUI-LOC.
3204
3205      *
3206      IF  SQLCODE NOT EQUAL 0
3207      AND SQLCODE NOT EQUAL 1403
3208      MOVE  "*** ERROR **  FETCHING FROM LOC_STAT_CURS CURSOR"
3209      TO WS-ERR-MSG
3210      PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3211      END-IF.
3212      *
3213      IF  SQLCODE EQUAL 1403
3214      MOVE 'Y' TO WS-EOC-LOC-STATE
3215      ELSE
3216      ADD 1 TO WS-STATE-COUNT
3217      EVALUATE WS-LOC-STATE
3218      WHEN 'NY'
3219      MOVE "Y" TO WS-NY-EXIST
3220      PERFORM 2627-SSL-CHECK THRU 2627-EXIT
3221      WHEN 'VA'
3222      MOVE "Y" TO WS-VA-EXIST
3223      WHEN 'MI'
3224      MOVE "Y" TO WS-MI-EXIST
3225      WHEN 'PA'
3226      MOVE "Y" TO WS-PA-EXIST
3227      WHEN 'NC'
3228      MOVE "Y" TO WS-NC-EXIST
3229      WHEN 'KY'
3230      MOVE "Y" TO WS-KY-EXIST
3231      WHEN 'WA'
3232      MOVE "Y" TO WS-WA-EXIST
3233      WHEN 'NJ'
3234      MOVE "Y" TO WS-NJ-EXIST
3235      END-EVALUATE
3236      END-IF.
3237
3238
3239      PERFORM 7302-DETERMINE-COVG THRU 7302-EXIT.
3240
3241      *
3242      2260-EXIT.
3243      *
3244      *-----*
3245      *  THIS PARAGRAPH WILL SELECT INFORMATION FROM THE INFC TO BE  *
3246      *  PRINTED ON THE SECOND PAGE OF THE POLICY OR ON THE REVISION. *
3247      *-----*
3248      *

```

```

3249 2300-GET-COVERAGE-INFO.
3250
3251
3252 *
3253 INITIALIZE
3254     WS-INFC-HSTP-GB-CODE-OF,
3255     WS-INFC-GB-ANNUAL-PX,
3256     WS-INFC-GB-WRITTEN-PX,
3257     WS-INFC-AU-COV-AU-LIAB,
3258     WS-INFC-AU-COV-AU-PIP,
3259     WS-INFC-AU-COV-AU-ADDED-PIP,
3260     WS-INFC-AU-COV-AU-MED-PAY,
3261     WS-INFC-AU-COV-AU-MED-EXP,
3262     WS-INFC-AU-COV-AU-UM,
3263     WS-INFC-AU-COV-AU-UI,
3264     WS-INFC-AU-COV-AU-COMP,
3265     WS-INFC-AU-COV-AU-SPECIFIED,
3266     WS-INFC-AU-COV-AU-COLLISION,
3267     WS-INFC-AU-COV-AU-TOW-LABOR,
3268     WS-INFC-AU-COV-AU-BUYBACK,
3269     WS-INFC-AU-COV-AU-RENTAL,
3270     WS-INFC-AU-COV-AU-PPI,
3271     WS-INFC-GB-EFF-DATE,
3272     WS-INFC-GB-EXP-DATE,
3273     WS-INFC-GB-END-NO,
3274     WS-INFC-GB-PRIMARY-STATE.
3275 *
3276 EXEC SQL AT ORA1
3277     SELECT INFC_HSTP_GB_CODE_OF,
3278            INFC_GB_ANNUAL_PX,
3279            INFC_GB_WRITTEN_PX,
3280            INFC_AU_COV_AU_LIAB,
3281            INFC_AU_COV_AU_PIP,
3282            INFC_AU_COV_AU_ADDED_PIP,
3283            INFC_AU_COV_AU_MED_PAY,
3284            INFC_AU_COV_AU_MED_EXP,
3285            INFC_AU_COV_AU_UM,
3286            INFC_AU_COV_AU_UI,
3287            INFC_AU_COV_AU_COMP,
3288            INFC_AU_COV_AU_SPECIFIED,
3289            INFC_AU_COV_AU_COLLISION,
3290            INFC_AU_COV_AU_TOW_LABOR,
3291            INFC_AU_COV_AU_BUYBACK,
3292            INFC_AU_COV_AU_RENTAL,
3293            INFC_AU_COV_AU_PPI,
3294            TO_CHAR(INFC_GB_EFF_DATE,'MM/DD/YYYY'),
3295            TO_CHAR(INFC_GB_EXP_DATE,'MM/DD/YYYY'),
3296            INFC_GB_END_NO,
3297     INFC_GB_PRIMARY_STATE
3298     INTO :WS-INFC-HSTP-GB-CODE-OF,
3299         :WS-INFC-GB-ANNUAL-PX,
3300         :WS-INFC-GB-WRITTEN-PX,
3301         :WS-INFC-AU-COV-AU-LIAB,
3302         :WS-INFC-AU-COV-AU-PIP,
3303         :WS-INFC-AU-COV-AU-ADDED-PIP,
3304         :WS-INFC-AU-COV-AU-MED-PAY,

```

```

3305         :WS-INFC-AU-COV-AU-MED-EXP,
3306         :WS-INFC-AU-COV-AU-UM,
3307         :WS-INFC-AU-COV-AU-UI,
3308         :WS-INFC-AU-COV-AU-COMP,
3309         :WS-INFC-AU-COV-AU-SPECIFIED,
3310         :WS-INFC-AU-COV-AU-COLLISION,
3311         :WS-INFC-AU-COV-AU-TOW-LABOR,
3312         :WS-INFC-AU-COV-AU-BUYBACK,
3313         :WS-INFC-AU-COV-AU-RENTAL,
3314         :WS-INFC-AU-COV-AU-PPI,
3315         :WS-INFC-GB-EFF-DATE,
3316         :WS-INFC-GB-EXP-DATE,
3317         :WS-INFC-GB-END-NO,
3318         :WS-INFC-GB-PRIMARY-STATE
3319     FROM INFC
3320     WHERE :WS-PPRT-INFC-GB-NO = INFC_GB_NO
3321 END-EXEC.
3322 *
3323 IF   SQLCODE NOT EQUAL 0
3324     MOVE "*** ERROR ** SELECTING INFC INFO - 2300"
3325     TO WS-ERR-MSG
3326     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3327     GO TO 2300-EXIT
3328 END-IF.
3329 *
3330 EVALUATE WS-INFC-HSTP-GB-CODE-OF
3331     WHEN 'NP'
3332         CONTINUE
3333     WHEN 'RV'
3334         CONTINUE
3335     WHEN OTHER
3336         PERFORM 2350-GET-ADD-COVERAGE-INFO THRU 2350-EXIT
3337 END-EVALUATE.
3338 *
3339 2300-EXIT.
3340 EXIT.
3341 *
3342 *-----*
3343 * THIS PARAGRAPH WILL SELECT TOTAL INFORMATION TO PRINT ON PAGE *
3344 * 2 OF THE POLICY. *
3345 *-----*
3346 *
3347 2350-GET-ADD-COVERAGE-INFO.
3348
3349 *
3350 * GET A SUM OF SINGLE LIMIT LIAB PREM
3351 *
3352     INITIALIZE
3353     WS-SUM-WRIT-PREM-LIAB.
3354 *
3355 EXEC SQL AT ORA1
3356     SELECT SUM(EXIT_GB_WRIT_PREM)
3357     INTO :WS-SUM-WRIT-PREM-LIAB
3358     FROM EXIT
3359     WHERE (EXIT_WC_CLASS_CODE
3360     IN

```



```

3361      ('ASPL','ASLL','SLL', 'PD', 'BI','SPL','SPLPD','RLOAN',
3362      'EXAPIP','MCSLL')
3363      OR  EXIT_AUCV_AU_NO IN
3364      (SELECT AUCV_AU_NO
3365      FROM AUTO_COVERAGE
3366      WHERE AUCV_AU_PLAN_NO IN ('2', '3', '4','7'))
3367      AND EXIT_GB_NO IN
3368      (SELECT COIT_EXIT_GB_NO
3369      FROM COIT
3370      WHERE COIT_INFC_GB_NO
3371      = :WS-PPRT-INFC-GB-NO)
3372  END-EXEC.
3373  *
3374  IF  SQLCODE NOT EQUAL 0
3375  AND SQLCODE NOT EQUAL 1403
3376  MOVE  "*** ERROR ** SELECTING SUM OF SLL PREM - 2350"
3377  TO WS-ERR-MSG
3378  PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3379  GO TO 2350-EXIT
3380  END-IF.
3381  *
3382  INITIALIZE
3383  WS-SUM-ASSM-GB-WRIT-AMOUNT.
3384  *
3385  EXEC SQL AT ORA1
3386  SELECT SUM(ASSM_GB_WRIT_AMOUNT)
3387  INTO :WS-SUM-ASSM-GB-WRIT-AMOUNT
3388  FROM STAS, ASSM, EXIT
3389  WHERE ASSM_EXIT_GB_NO  = EXIT_GB_NO
3390  AND ASSM_INFC_GB_NO  = :WS-PPRT-INFC-GB-NO
3391  AND STAS_GB_NO      = ASSM_STAS_GB_NO
3392  AND STAS_GB_PRINT_LEVEL = 'EP'
3393  AND (EXIT_WC_CLASS_CODE
3394  IN
3395  ('ASPL','ASLL','SLL', 'PD', 'BI','SPL','SPLPD','RLOAN'
3396  , 'MCSLL')
3397  OR  EXIT_AUCV_AU_NO IN
3398  (SELECT AUCV_AU_NO
3399  FROM AUTO_COVERAGE
3400  WHERE AUCV_AU_PLAN_NO IN ('2', '3', '4','7'))
3401  AND EXIT_GB_NO IN
3402  (SELECT COIT_EXIT_GB_NO
3403  FROM COIT
3404  WHERE COIT_INFC_GB_NO
3405  = :WS-PPRT-INFC-GB-NO)
3406  END-EXEC.
3407  *
3408  IF  SQLCODE NOT EQUAL 0
3409  AND SQLCODE NOT EQUAL 1403
3410  MOVE  "*** ERROR ** SELECTING SUM OF SLL PREM - 2350"
3411  TO WS-ERR-MSG
3412  PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3413  GO TO 2350-EXIT
3414  END-IF.
3415  *
3416  COMPUTE WS-SUM-WRIT-PREM-LIAB = WS-SUM-WRIT-PREM-LIAB

```

```

3417             + WS-SUM-ASSM-GB-WRIT-AMOUNT.
3418 *
3419 * GET SUM OF HIRED PHYSICAL DAMAGE PREM
3420 *
3421     INITIALIZE
3422         WS-SUM-WRIT-PREM-HIRPHY
3423 *
3424     EXEC SQL AT ORA1
3425         SELECT SUM(EXIT_GB_WRIT_PREM)
3426             INTO :WS-SUM-WRIT-PREM-HIRPHY
3427             FROM EXIT
3428             WHERE EXIT_GB_NO IN
3429                 (SELECT COIT_EXIT_GB_NO
3430                  FROM COIT
3431                  WHERE COIT_INFC_GB_NO
3432                     = :WS-PPRT-INFC-GB-NO)
3433             AND EXIT_AUCV_AU_NO IN
3434                 (SELECT AUCV_AU_NO
3435                  FROM AUTO_COVERAGE
3436                  WHERE AUCV_AU_PLAN_NO = '5')
3437             AND EXIT_WC_CLASS_CODE NOT IN ('GKCOLL', 'GKCOMP')
3438     END-EXEC.
3439 *
3440     IF     SQLCODE NOT EQUAL 0
3441         AND SQLCODE NOT EQUAL 1403
3442     MOVE "*** ERROR ** SELECTING SUM OF HIRED PHY DAMAGE - 2350"
3443         TO WS-ERR-MSG
3444         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3445         GO TO 2350-EXIT
3446     END-IF.
3447 *
3448 *
3449     INITIALIZE
3450         WS-SUM-ASSM-GB-WRIT-AMOUNT.
3451 *
3452     EXEC SQL AT ORA1
3453         SELECT SUM(ASSM_GB_WRIT_AMOUNT)
3454             INTO :WS-SUM-ASSM-GB-WRIT-AMOUNT
3455             FROM STAS, ASSM, EXIT
3456             WHERE ASSM_EXIT_GB_NO   = EXIT_GB_NO
3457             AND ASSM_INFC_GB_NO     = :WS-PPRT-INFC-GB-NO
3458             AND STAS_GB_NO          = ASSM_STAS_GB_NO
3459             AND STAS_GB_PRINT_LEVEL = 'EP'
3460             AND EXIT_GB_NO IN
3461                 (SELECT COIT_EXIT_GB_NO
3462                  FROM COIT
3463                  WHERE COIT_INFC_GB_NO
3464                     = :WS-PPRT-INFC-GB-NO)
3465             AND EXIT_AUCV_AU_NO IN
3466                 (SELECT AUCV_AU_NO
3467                  FROM AUTO_COVERAGE
3468                  WHERE AUCV_AU_PLAN_NO = '5')
3469     END-EXEC.
3470 *
3471     IF     SQLCODE NOT EQUAL 0
3472         AND SQLCODE NOT EQUAL 1403

```

```

3473         MOVE "*** ERROR ** SELECTING SUM OF SLL PREM - 2350"
3474             TO WS-ERR-MSG
3475             PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3476             GO TO 2350-EXIT
3477     END-IF.
3478 *
3479     COMPUTE WS-SUM-WRIT-PREM-HIRPHY = WS-SUM-WRIT-PREM-HIRPHY
3480             + WS-SUM-ASSM-GB-WRIT-AMOUNT.
3481 *
3482     INITIALIZE
3483         WS-SUM-WRIT-PREM-HIRPHY-1403.
3484
3485 *
3486     EXEC SQL AT ORA1
3487         SELECT DECODE(EXIT_WC_CLASS_CODE,
3488             'MCSLL', 'SLL',
3489             'ASLL', 'SLL',
3490             'ASPL', 'SLL',
3491             'IPIP', 'PIP',
3492             'APIP', 'PIP',
3493             'AAPIP', 'PIP',
3494             'PI PST', 'PIP',
3495             'PIPI', 'PIP',
3496             'GPIP', 'PIP',
3497             'AUM', 'UM',
3498             'AUMPD', 'UMPD',
3499             'AUI', 'UI',
3500             'UICC', 'UI',
3501             'AMED', 'MED',
3502             'ILB', 'MED',
3503             'ACOLL', 'COLL',
3504             'COLWG', 'COLL',
3505             'ACOMP', 'COMP',
3506             'CMPWG', 'COMP',
3507             'A6601', '6601',
3508             'NFUM', 'UM',
3509             'NNFUM', 'UM',
3510             'UMS', 'UM',
3511             'UMN', 'UM',
3512             'UMBI', 'UM',
3513             'UMBIPD', 'UM',
3514             'UMIN', 'UM',
3515             'UMOT', 'UM',
3516             'UMINPD', 'UM',
3517             'UMOTPD', 'UM',
3518             'UMUIIN', 'UM',
3519             'UMUIOT', 'UM',
3520             'UIS', 'UI',
3521             'UIN', 'UI',
3522             'UIOT', 'UI',
3523             'UIIN', 'UI',
3524             'UIBIPD', 'UI',
3525             'UIPDOT', 'UI',
3526             'UIPDIN', 'UI',
3527             'UICCIN', 'UI',
3528             'UICCOT', 'UI',

```

```

3529          'CPIP', 'PIP',
3530          'EXMED', 'PIP',
3531      'UMSIN', 'UM',
3532          'UMSOT', 'UM',
3533          'UMNIN', 'UM',
3534          'UMNOT', 'UM',
3535          'UISIN', 'UI',
3536          'UISOT', 'UI',
3537          'UININ', 'UI',
3538          'UINOT', 'UI',
3539          'NOUM', 'UM',
3540      'NOUI', 'UI',
3541      'HUM', 'UM',
3542      'HUI', 'UI',
3543      'PIPF', 'PIP',
3544          'PIPO', 'PIP',
3545          'PEDPIP', 'PIP',
3546          'PIPOB', 'PIP',
3547          'OBWRK', 'PIP',
3548          'OBAE', 'PIP',
3549          'OBFE', 'PIP',
3550          'OBDB', 'PIP',
3551          'SPL', 'BI',
3552      'SPLPD', 'PD',
3553      'UUBN', 'UM',
3554      'UUBPN', 'UM',
3555      'UUBS', 'UM',
3556      'UUBPS', 'UM',
3557          'UMINAO', 'UM',
3558          'UMOTAO', 'UM',
3559          'UMINRB', 'UM',
3560          'UMOTRB', 'UM',
3561          EXIT_WC_CLASS_CODE, EXIT_WC_CLASS_CODE)
3562 FROM EXIT
3563 WHERE EXIT_GB_NO IN
3564     (SELECT COIT_EXIT_GB_NO
3565     FROM COIT
3566     WHERE COIT_INFC_GB_NO
3567     = :WS-PPRT-INFC-GB-NO)
3568 AND EXIT_AUCV_AU_NO IN
3569     (SELECT AUCV_AU_NO
3570     FROM AUTO_COVERAGE
3571     WHERE AUCV_AU_PLAN_NO = '5')
3572 AND ROWNUM = 1
3573 END-EXEC.
3574 *
3575 IF     SQLCODE NOT EQUAL 0
3576     AND SQLCODE NOT EQUAL 1403
3577     MOVE "*** ERROR ** SELECTING SUM OF HIRED PHY DAMAGE - 2350"
3578     TO WS-ERR-MSG
3579     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3580     GO TO 2350-EXIT
3581 END-IF.
3582 *
3583 IF SQLCODE EQUAL 1403
3584     MOVE 'Y' TO WS-SUM-WRIT-PREM-HIRPHY-1403

```

```

3585     ELSE
3586         MOVE 'N' TO WS-SUM-WRIT-PREM-HIRPHY-1403
3587     END-IF.
3588 *
3589     INITIALIZE
3590         WS-SUM-ASSM-MVFEE.
3591 *
3592     EXEC SQL AT ORA1
3593         SELECT ASSM_GB_ASSMNT_CODE,
3594             SUM(ASSM_GB_AMOUNT)
3595             INTO :WS-ASSM-CODE-MVFEE,
3596                 :WS-SUM-ASSM-MVFEE
3597         FROM ASSM
3598         WHERE :WS-PPRT-INFC-GB-NO = ASSM_INFC_GB_NO
3599             AND ASSM_GB_ASSMNT_CODE = 'MVFEE'
3600         GROUP BY ASSM_GB_ASSMNT_CODE
3601     END-EXEC.
3602 *
3603     IF SQLCODE NOT EQUAL 0
3604         AND SQLCODE NOT EQUAL 1403
3605         MOVE "*** ERROR ** SELECTING MVFEE ASSM INFO - 2350"
3606         TO WS-ERR-MSG
3607         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3608     END-IF.
3609 *
3610     IF SQLCODE EQUAL 1403 THEN
3611         MOVE 'Y' TO WS-SUM-ASSM-MVFEE-1403
3612     ELSE
3613         MOVE 'N' TO WS-SUM-ASSM-MVFEE-1403
3614     END-IF.
3615 *
3616     2350-EXIT.
3617     EXIT.
3618 *
3619 *-----*
3620 * THIS PARAGRAPH WILL RETRIEVE THE ASSESSMENTS BASED ON WHETHER *
3621 * THEY ARE COVERAGES, SURCHARGES, OR MODIFICATIONS TO THE POLICY *
3622 *-----*
3623     2390-GET-POLICY-ASSM-INFO.
3624
3625 *
3626     INITIALIZE WS-ASSM-GB-AMOUNT-POLY,
3627         WS-ASSM-TABLE,
3628         WS-KY-ASSM-TABLE.
3629 *
3630     EXEC SQL AT ORA1
3631         SELECT DECODE(ASSM_GB_ASSMNT_CODE,'SIZMOD','EXPRED',
3632             ASSM_GB_ASSMNT_CODE),
3633             DECODE(ASSM_GB_ASSMNT_CODE,
3634                 'SIZMOD','EXPENSE MODIFICATION',
3635                 'EXPRED','EXPENSE MODIFICATION',
3636                 ASDC_GB_DESC),
3637             LOCT_STAT_GB_ALPHA_STATE_IN,
3638             SUM(NVL(ASSM_GB_WRIT_AMOUNT,0))
3639         INTO :WS-ASSM-GB-ASSMNT-CODE-2A,
3640             :WS-ASDC-GB-DESC-2A,

```



```

3641         :WS-ASSM-LOCT-2A,
3642         :WS-ASSM-GB-WRIT-AMOUNT-2A
3643     FROM STAS, ASSM, ASDC, LOC, MCTB
3644     WHERE ASSM_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
3645     AND ASSM_GB_ASSMNT_CODE != 'MVFEE'
3646     AND ASSM_GB_ASSMNT_CODE != 'SUCITY'
3647     AND STAS_GB_NO(+) = ASSM_STAS_GB_NO
3648     AND ASSM_GB_ASSMNT_CODE = ASDC_GB_CODE
3649     AND MCTB_GB_CODE = ASSM_GB_ASSMNT_CODE
3650     AND MCTB_GB_CODE_TYPE = 'AUPRT2A'
3651     AND MCTB_GB_EDIT = :WS-MCTB-EDIT
3652     AND ASSM_LOCT_GB_NO = LOCT_GB_NO
3653     AND (ASSM_STAS_GB_NO IS NULL
3654         OR NVL(STAS_GB_PRINT_LEVEL,'P') = 'P')
3655     GROUP BY ASSM_GB_ASSMNT_CODE,
3656             ASDC_GB_DESC,
3657             LOCT_STAT_GB_ALPHA_STATE_IN
3658     ORDER BY ASSM_GB_ASSMNT_CODE
3659 END-EXEC.
3660
3661 *
3662     PERFORM VARYING WS-WHAT-SUB
3663     FROM 1 BY 1 UNTIL WS-ASSM-GB-ASSMNT-CODE-2A(WS-WHAT-SUB)
3664     EQUAL SPACES
3665 *
3666     IF WS-MCTB-EDIT = 'S' THEN
3667         MOVE SPACES TO WS-HOLD-DESC
3668         STRING WS-ASSM-LOCT-2A(WS-WHAT-SUB)
3669         DELIMITED BY " ",
3670         " "
3671         DELIMITED BY SIZE,
3672         WS-ASDC-GB-DESC-2A(WS-WHAT-SUB)
3673         DELIMITED BY SIZE,
3674         INTO WS-HOLD-DESC
3675         MOVE WS-HOLD-DESC TO WS-ASDC-GB-DESC(WS-WHAT-SUB)
3676         ADD WS-ASSM-GB-WRIT-AMOUNT-2A(WS-WHAT-SUB)
3677         TO WS-ASSM-GB-AMOUNT-POLY
3678         MOVE WS-ASSM-GB-WRIT-AMOUNT-2A(WS-WHAT-SUB)
3679         TO WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB)
3680     END-IF
3681     ADD 1 TO WS-COUNT-ASSM
3682 *
3683     IF WS-MCTB-EDIT = 'M' THEN
3684         IF WS-ASSM-GB-ASSMNT-CODE-2A(WS-WHAT-SUB) =
3685             WS-ASSM-HOLD THEN
3686             ADD WS-ASSM-GB-WRIT-AMOUNT-2A(WS-WHAT-SUB)
3687             TO WS-ASSM-REDUCT-WRIT-AMOUNT(WS-REDUCT)
3688             SUBTRACT 1 FROM WS-COUNT-ASSM
3689         ELSE
3690             ADD 1 TO WS-REDUCT
3691             MOVE WS-ASDC-GB-DESC-2A(WS-WHAT-SUB)
3692             TO WS-ASSM-REDUCT-DESC(WS-REDUCT)
3693             MOVE WS-ASSM-GB-WRIT-AMOUNT-2A(WS-WHAT-SUB)
3694             TO WS-ASSM-REDUCT-WRIT-AMOUNT(WS-REDUCT)
3695             MOVE WS-ASSM-GB-ASSMNT-CODE-2A(WS-WHAT-SUB)
3696             TO WS-ASSM-HOLD
3697         END-IF

```

```

3697         END-IF
3698     END-PERFORM.
3699 *
3700     IF WS-MCTB-EDIT = 'S' THEN
3701         IF WS-SUM-ASSM-MVFEE-1403 = "N" THEN
3702             ADD WS-SUM-ASSM-MVFEE TO WS-ASSM-GB-AMOUNT-POLY
3703         END-IF
3704 *
3705         INITIALIZE WS-KY-ASSM-TABLE
3706 *
3707         EXEC SQL AT ORA1
3708             SELECT ASSM_GB_ASSMNT_CODE,
3709                 ASDC_GB_DESC,
3710                 LOCT_STAT_GB_ALPHA_STATE_IN,
3711                 SUM(NVL(ASSM_GB_WRIT_AMOUNT,0)),
3712                 NVL(CYHS.CYCY_GB_CODE,LOC.CYCY_GB_CODE)
3713             INTO :WS-ASSM-GB-ASSMNT-CODE-KY,
3714                 :WS-ASDC-GB-DESC-KY,
3715                 :WS-ASSM-LOCT-KY,
3716                 :WS-ASSM-GB-WRIT-AMOUNT-KY,
3717                 :WS-CYCY-GB-CODE-KY
3718             FROM STAS, ASSM, ASDC, LOC, MCTB, CYHS
3719             WHERE ASSM_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
3720                 AND ASSM_GB_ASSMNT_CODE = 'SUCITY'
3721                 AND STAS_GB_NO(+) = ASSM_STAS_GB_NO
3722                 AND ASSM_GB_ASSMNT_CODE = ASDC_GB_CODE
3723                 AND MCTB_GB_CODE = ASSM_GB_ASSMNT_CODE
3724                 AND MCTB_GB_CODE_TYPE = 'AUPRT2A'
3725                 AND MCTB_GB_EDIT = :WS-MCTB-EDIT
3726                 AND ASSM_LOCT_GB_NO = LOC.LOCT_GB_NO
3727                 AND CYHS.LOCT_GB_NO(+) = LOC.LOCT_GB_NO
3728                 AND NVL(CYHS.CYCY_GB_CODE,LOC.CYCY_GB_CODE) != 'NA'
3729                 AND (ASSM_STAS_GB_NO IS NULL
3730                     OR NVL(STAS_GB_PRINT_LEVEL,'P') = 'P')
3731         GROUP BY ASSM_GB_ASSMNT_CODE,
3732                 ASDC_GB_DESC,
3733                 LOCT_STAT_GB_ALPHA_STATE_IN,
3734                 NVL(CYHS.CYCY_GB_CODE,LOC.CYCY_GB_CODE)
3735         ORDER BY ASSM_GB_ASSMNT_CODE
3736     END-EXEC
3737 *
3738     PERFORM VARYING WS-KY-SUB
3739         FROM 1 BY 1 UNTIL
3740             WS-ASSM-GB-ASSMNT-CODE-KY(WS-KY-SUB) EQUAL SPACES
3741 *
3742         IF WS-MCTB-EDIT = 'S' THEN
3743             MOVE SPACES TO WS-HOLD-DESC
3744             STRING WS-ASSM-LOCT-KY(WS-KY-SUB)
3745                 DELIMITED BY " ",
3746                 " "
3747             DELIMITED BY SIZE,
3748             WS-ASDC-GB-DESC-KY(WS-KY-SUB)
3749             DELIMITED BY SIZE,
3750             INTO WS-HOLD-DESC
3751             MOVE WS-HOLD-DESC TO WS-ASDC-GB-DESC(WS-WHAT-SUB)
3752             MOVE WS-CYCY-GB-CODE-KY(WS-KY-SUB)

```

```

3753         TO WS-CYCY-CDE-KY (WS-WHAT-SUB)
3754     ADD WS-ASSM-GB-WRIT-AMOUNT-KY(WS-KY-SUB)
3755     TO WS-ASSM-GB-AMOUNT-POLY
3756     MOVE WS-ASSM-GB-WRIT-AMOUNT-KY(WS-KY-SUB)
3757     TO WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB)
3758     ADD 1 TO WS-COUNT-ASSM
3759     END-IF
3760     ADD 1 TO WS-WHAT-SUB
3761     END-PERFORM
3762 *
3763     INITIALIZE WS-ASSM-TABLE
3764 *
3765     EXEC SQL AT ORA1
3766         SELECT ASSM_GB_ASSMNT_CODE,
3767             ASDC_GB_DESC,
3768             LOCT_STAT_GB_ALPHA_STATE_IN,
3769             SUM(NVL(ASSM_GB_WRIT_AMOUNT,0))
3770     INTO   :WS-ASSM-GB-ASSMNT-CODE-2A,
3771           :WS-ASDC-GB-DESC-2A,
3772           :WS-ASSM-LOCT-2A,
3773           :WS-ASSM-GB-WRIT-AMOUNT-2A
3774     FROM STAS, ASSM, ASDC, LOC
3775     WHERE ASSM_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
3776     AND ASSM_GB_ASSMNT_CODE != 'MVFEE'
3777     AND ASSM_GB_ASSMNT_CODE != 'SUCITY'
3778     AND STAS_GB_NO(+) = ASSM_STAS_GB_NO
3779     AND ASSM_GB_ASSMNT_CODE = ASDC_GB_CODE
3780     AND ASSM_LOCT_GB_NO = LOCT_GB_NO
3781     AND (ASSM_STAS_GB_NO IS NULL
3782         OR NVL(STAS_GB_PRINT_LEVEL,'P') = 'P')
3783     AND ASSM_GB_ASSMNT_CODE NOT IN
3784         (SELECT MCTB_GB_CODE FROM MCTB
3785          WHERE MCTB_GB_CODE_TYPE = 'AUPRT2A'
3786            AND MCTB_GB_EDIT IN('C','S','M'))
3787     GROUP BY ASSM_GB_ASSMNT_CODE,
3788             ASDC_GB_DESC,
3789             LOCT_STAT_GB_ALPHA_STATE_IN
3790     ORDER BY ASSM_GB_ASSMNT_CODE
3791     END-EXEC
3792 *
3793     PERFORM VARYING WS-NEXT-SUB
3794     FROM 1 BY 1 UNTIL WS-ASSM-GB-ASSMNT-CODE-2A(WS-NEXT-SUB)
3795     EQUAL SPACES
3796     IF WS-MCTB-EDIT = 'S' THEN
3797         MOVE SPACES TO WS-HOLD-DESC
3798         STRING WS-ASSM-LOCT-2A(WS-NEXT-SUB)
3799         DELIMITED BY " ",
3800         " "
3801         DELIMITED BY SIZE,
3802         WS-ASDC-GB-DESC-2A(WS-NEXT-SUB)
3803         DELIMITED BY SIZE,
3804         INTO WS-HOLD-DESC
3805         MOVE WS-HOLD-DESC TO WS-ASDC-GB-DESC(WS-WHAT-SUB)
3806         ADD WS-ASSM-GB-WRIT-AMOUNT-2A(WS-NEXT-SUB)
3807         TO WS-ASSM-GB-AMOUNT-POLY
3808         MOVE WS-ASSM-GB-WRIT-AMOUNT-2A(WS-NEXT-SUB)

```

```

3809             TO WS-ASSM-GB-WRIT-AMT(W5-WHAT-SUB)
3810         END-IF
3811         ADD 1 TO WS-COUNT-ASSM
3812         ADD 1 TO WS-WHAT-SUB
3813     END-PERFORM
3814 END-IF.
3815 *
3816 2390-EXIT.
3817 EXIT.
3818 *
3819 *
3820 2395-FETCH-OTHER-COV.
3821 *
3822     INITIALIZE
3823     WS-ASSM-CODE,
3824     WS-ASSM-DESC,
3825     WS-ASSM-WRIT.
3826 *
3827 EXEC SQL
3828     FETCH OTHER_COV_CURS
3829     INTO :WS-ASSM-CODE,
3830         :WS-ASSM-DESC,
3831         :WS-ASSM-WRIT
3832 END-EXEC.
3833
3834     IF WS-ASSM-DESC =
3835         'GARAGEKEEPERS INSURANCE - COLLISION COVERAGE'
3836         MOVE 'GARAGEKEEPERS COVERAGE - COLL' TO WS-ASSM-DESC
3837     END-IF.
3838
3839     IF WS-ASSM-DESC =
3840         'GARAGEKEEPERS INSURANCE - COMPREHENSIVE COVERAGE'
3841         MOVE 'GARAGEKEEPERS COVERAGE - COMP' TO WS-ASSM-DESC
3842     END-IF.
3843
3844
3845
3846
3847
3848 *
3849 IF     SQLCODE NOT EQUAL 0
3850 AND SQLCODE NOT EQUAL 1403
3851 MOVE  "*** ERROR **  FETCHING FROM OTHER_COV_CURS CURSOR"
3852 TO WS-ERR-MSG
3853 PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3854 END-IF.
3855 *
3856 IF SQLCODE EQUAL 1403
3857     MOVE 'Y' TO WS-EOC-OTHER-COV
3858 END-IF.
3859 *
3860 2395-EXIT.
3861 *
3862 *
3863 2396-PROCESS-OTHER-COV.
3864 *
```

```

3865 IF WS-PG2-LINECNT >= 39 THEN
3866     PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
3867 END-IF.
3868 MOVE WS-ASSM-DESC TO
3869     PRT-COVERAGE-DESC
3870 IF WS-ASSM-DESC EQUAL
3871     'TRAILER INTERCHANGE'
3872     MOVE " (SEE ENDORSEMENT)"
3873     TO PRT-COVERAGE-SEP-STATED
3874 END-IF
3875 IF WS-ASSM-WRIT > 0 THEN
3876     MOVE WS-ASSM-WRIT
3877     TO PRT-COST-DECIMAL
3878 ELSE
3879     MOVE WS-ASSM-WRIT
3880     TO PRT-ZERO-COST-DECIMAL
3881 END-IF.
3882 WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
3883 ADD 2 TO WS-PG2-LINECNT.
3884 *
3885     PERFORM 2395-FETCH-OTHER-COV THRU 2395-EXIT.
3886
3887 *
3888 2396-EXIT.
3889 *
3890 *-----*
3891 * THIS PARAGRAPH WILL PERFORM PARAGRAPHS THAT SELECTS INFORMATION*
3892 * FROM THE DATABASE TO BE PRINTED ON THE SCHEDULE OF AUTOS, LOSS *
3893 * PAYEE LIST, AND NOTICE OF LOSS PAYEES. *
3894 *-----*
3895 *
3896 2400-GET-ADD-POLICY-INFO.
3897
3898 *
3899 IF WS-INFC-HSTP-GB-CODE-OF NOT EQUAL 'RV'
3900 AND WS-INFC-HSTP-GB-CODE-OF NOT EQUAL 'NP'
3901     PERFORM 2900-GET-HIRED-NON-OWNED-INFO THRU 2900-EXIT
3902 END-IF.
3903 *
3904 INITIALIZE
3905     WS-ENDORSEMENT-TABLE.
3906 *
3907 MOVE 'N' TO WS-EOC-ATTACH-CURS.
3908 MOVE 1 TO WS-ENTRY-COUNT.
3909 MOVE 1 TO WS-ENDORSEMENT-COUNT.
3910 *
3911 IF WS-INFC-HSTP-GB-CODE-OF NOT EQUAL 'RV'
3912 AND WS-INFC-HSTP-GB-CODE-OF NOT EQUAL 'NP'
3913     PERFORM 2525-OPEN-ATTACH-CURS THRU 2525-EXIT
3914     PERFORM 2550-FETCH-ATTACH-CURS THRU 2550-EXIT
3915     PERFORM 2500-GET-END-INFO THRU 2500-EXIT
3916     UNTIL WS-EOC-ATTACH-CURS = 'Y'
3917 END-IF.
3918 *
3919 PERFORM 2600-GET-AUTO-INFO THRU 2600-EXIT.
3920 *

```

```

3921 2400-EXIT.
3922 EXIT.
3923 *
3924 * -----*
3925 * THIS PARAGRAPH WILL SELECT INFORMATION FROM THE ENFM TABLE FOR *
3926 * THE ATTACHED ENDORSEMENTS TO BE PRINTED ON PAGE 3 ITEM SIX OF *
3927 * THIS AUTO POLICY. *
3928 * -----*
3929 *
3930 2500-GET-END-INFO.
3931 *
3932 EXEC SQL AT ORA1
3933 SELECT RPAD(ENFM_EDDS_GB_FORM_NO,20,'%'),
3934 LENGTH(ENFM_EDDS_GB_FORM_NO),
3935 ENFM_GB_EDITION
3936 INTO :WS-ENDFORM,
3937 :WS-ENDFORM-LEN,
3938 :WS-EDITION-DATE
3939 FROM ENFM
3940 WHERE ENFM_GB_NO = :WS-ATCH-ENFM-GB-NO
3941 END-EXEC.
3942 *
3943 IF SQLCODE NOT EQUAL 0
3944 MOVE "*** ERROR ** SELECTING ENDORSEMENT INFO - 2500"
3945 TO WS-ERR-MSG
3946 PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3947 END-IF.
3948 *
3949 MOVE SPACES TO WS-PRT-ENDFORM.
3950 STRING WS-ENDFORM
3951 DELIMITED BY "%",
3952 "("
3953 DELIMITED BY SIZE,
3954 WS-EDITION-DATE
3955 DELIMITED BY SIZE,
3956 ")"
3957 DELIMITED BY SIZE,
3958 INTO WS-PRT-ENDFORM.
3959 *
3960 EVALUATE WS-ENDORSEMENT-COUNT
3961 WHEN 1
3962 MOVE WS-PRT-ENDFORM TO WS-END-1(WS-ENTRY-COUNT)
3963 ADD 1 TO WS-ENDORSEMENT-COUNT
3964 WHEN 2
3965 MOVE WS-PRT-ENDFORM TO WS-END-2(WS-ENTRY-COUNT)
3966 ADD 1 TO WS-ENDORSEMENT-COUNT
3967 WHEN 3
3968 MOVE WS-PRT-ENDFORM TO WS-END-3(WS-ENTRY-COUNT)
3969 ADD 1 TO WS-ENDORSEMENT-COUNT
3970 WHEN 4
3971 MOVE WS-PRT-ENDFORM TO WS-END-4(WS-ENTRY-COUNT)
3972 MOVE 1 TO WS-ENDORSEMENT-COUNT
3973 ADD 1 TO WS-ENTRY-COUNT
3974 END-EVALUATE.
3975 *
3976 IF WS-ENTRY-COUNT-3 = 200

```



```

3977         MOVE "*** ERROR ** ENDORSEMENT TABLE SIZE EXCEEDED."
3978         TO WS-ERR-MSG
3979         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
3980     END-IF;
3981 *
3982     PERFORM 2550-FETCH-ATTACH-CURS THRU 2550-EXIT.
3983 *
3984     2500-EXIT.
3985     EXIT.
3986 *
3987 * -----*
3988 *   THIS PARAGRAPH WILL OPEN THE ATTACH_CURS.           *
3989 * -----*
3990 *
3991     2525-OPEN-ATTACH-CURS.
3992 *
3993     EXEC SQL
3994         OPEN ATTACH_CURS
3995     END-EXEC.
3996 *
3997     IF   SQLCODE NOT EQUAL 0
3998         AND SQLCODE NOT EQUAL 1403
3999         MOVE "*** ERROR ** OPENING ATTACH_CURS CURSOR"
4000         TO WS-ERR-MSG
4001         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4002     END-IF.
4003 *
4004     IF SQLCODE EQUAL 1403
4005         MOVE 'Y' TO WS-EOC-ATTACH-CURS
4006     END-IF.
4007 *
4008     2525-EXIT.
4009     EXIT.
4010 *
4011 * -----*
4012 *   THIS PARAGRAPH WILL FETCH FROM THE ATTACH_CURS.       *
4013 * -----*
4014 *
4015     2550-FETCH-ATTACH-CURS.
4016 *
4017     EXEC SQL
4018         FETCH ATTACH_CURS
4019         INTO :WS-ATCH-ENFM-GB-NO
4020     END-EXEC.
4021 *
4022     IF   SQLCODE NOT EQUAL 0
4023         AND SQLCODE NOT EQUAL 1403
4024         MOVE "*** ERROR ** FETCHING FROM ATTACH_CURS CURSOR"
4025         TO WS-ERR-MSG
4026         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4027     END-IF.
4028 *
4029     IF SQLCODE EQUAL 1403
4030         MOVE 'Y' TO WS-EOC-ATTACH-CURS
4031     END-IF.
4032 *

```

```

4033 2550-EXIT.
4034 EXIT.
4035 *
4036 * -----*
4037 * THIS PARAGRAPH WILL INITIALIZE THE WS-SCHEDULE-AUTO-TABLE AND *
4038 * THE WS-SCHEDULE-PREM-TABLE, OPEN THE AUTO_CURS, FETCH FROM THE *
4039 * AUTO_CURS AND PERFORM A PARAGRAPH WHICH PROCESS THE AUTO INFO. *
4040 * -----*
4041 *
4042 2600-GET-AUTO-INFO.
4043 *
4044     MOVE 1 TO WS-ENTRY-COUNT.
4045     MOVE 'N' TO WS-EOC-AUTO.
4046 *
4047     INITIALIZE
4048         WS-EXPOSURE-EXIST-TABLE
4049         WS-SCHEDULE-AUTO-TABLE
4050         WS-SCHEDULE-NET-PREM-TABLE
4051         WS-SCHEDULE-WRIT-PREM-TABLE
4052         WS-HOLD-STATE.
4053 *
4054 * IF THE HSTP IS EQUAL TO 'RV' OR 'NP', WE WILL NEED THE PREVIOUS
4055 * INFORCE NUMBER. IF THE HSTP IS 'RV', THE PREVIOUS INFORCE WILL BE
4056 * USED BY THE AUTO_CURS TO SELECT VEHICLES WHICH WERE MODIFIED IN THIS
4057 * ENDORSEMENT. IF THE HSTP IS 'NP', THE PREVIOUS INFORCE WILL BE
4058 * MOVED TO THE CURRENT INFORCE. SINCE, THE RATER CAN MODIFIED VESTED
4059 * INTEREST ON A 'NP', THE PREVIOUS INFORCE MUST BE USED TO SELECT THE
4060 * VEHICLES BECAUSE NO VEHICLES ARE ATTACHED TO A 'NP' ENDORSEMENT.
4061 * HOWEVER, VESTED INTEREST ARE ATTACHED TO 'NP' AND ONLY THE VEHICLES
4062 * WHICH HAVE A VESTED INTERESTED MODIFIED WILL BE PRINTED.
4063 *
4064     IF WS-INFC-HSTP-GB-CODE-OF = 'RV'
4065     OR WS-INFC-HSTP-GB-CODE-OF = 'NP'
4066     PERFORM 4300-GET-PREV-INFC THRU 4300-EXIT
4067     MOVE WS-PPRT-INFC-GB-NO TO WS-HOLD-PPRT-INFC-GB-NO
4068     MOVE WS-PREV-INFC-GB-NO TO WS-HOLD-PREV-INFC-GB-NO
4069     IF WS-INFC-HSTP-GB-CODE-OF = 'NP'
4070     MOVE WS-PREV-INFC-GB-NO TO WS-PPRT-INFC-GB-NO
4071     MOVE 0 TO WS-PREV-INFC-GB-NO
4072     END-IF
4073     ELSE
4074     MOVE 0 TO WS-PREV-INFC-GB-NO
4075     MOVE 0 TO WS-HOLD-PPRT-INFC-GB-NO
4076     MOVE 0 TO WS-HOLD-PREV-INFC-GB-NO
4077     END-IF.
4078 *
4079     EXEC SQL
4080         OPEN AUTO_CURS
4081     END-EXEC.
4082 *
4083     IF SQLCODE NOT EQUAL 0
4084     AND SQLCODE NOT EQUAL 1403
4085     MOVE "*** ERROR ** OPENING AUTO_CURS CUROSR"
4086     TO WS-ERR-MSG
4087     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4088     END-IF.

```

```

4089 *
4090 IF WS-INFC-HSTP-GB-CODE-OF = 'NP'
4091     MOVE WS-HOLD-PREV-INFC-GB-NO TO WS-PREV-INFC-GB-NO
4092 END-IF.
4093 *
4094 PERFORM 2650-FETCH-AUTO-INFO THRU 2650-EXIT.
4095 *
4096 PERFORM 2625-PROC-AUTO-INFO THRU 2625-EXIT
4097     UNTIL WS-EOC-AUTO = 'Y'.
4098 *
4099 IF WS-INFC-HSTP-GB-CODE-OF = 'RV'
4100     OR WS-INFC-HSTP-GB-CODE-OF = 'NP'
4101     MOVE WS-HOLD-PPRT-INFC-GB-NO TO WS-PPRT-INFC-GB-NO
4102 END-IF.
4103 *
4104 * AAP ADDED TO GET HCOLL INFO WHEN PA LOC EXISTS
4105 *
4106 IF WS-PA-EXIST = "Y" THEN
4107     EXEC SQL AT ORA1
4108         SELECT DEDUCT,
4109             LIMIT,
4110             DEDUCT_LEN,
4111             LIMIT_LEN
4112         INTO WS-HCOLL-DEDUCT,
4113             WS-HCOLL-LIMIT,
4114             WS-DEDUCT-LEN,
4115             WS-LIMIT-LEN
4116         FROM (SELECT TO_CHAR(EXIT_GB_DEDUCTIBLE) DEDUCT,
4117             AUCV_AU_LIMIT LIMIT,
4118             LENGTH(TO_CHAR(EXIT_GB_DEDUCTIBLE)) DEDUCT_LEN,
4119             LENGTH(AUCV_AU_LIMIT) LIMIT_LEN
4120             FROM AUCV, EXIT
4121             WHERE AUCV_AU_NO = EXIT_AUCV_AU_NO
4122                 AND EXIT_GB_NO IN (SELECT COIT_EXIT_GB_NO
4123                     FROM COIT
4124                     WHERE COIT_INFC_GB_NO
4125                         = :WS-PPRT-INFC-GB-NO)
4126                 AND AUCV_AU_TYPE = 'HCOLL'
4127             ORDER BY LIMIT, DEDUCT DESC)
4128         WHERE ROWNUM = 1
4129     END-EXEC
4130 END-IF.
4131 *
4132 2600-EXIT.
4133 EXIT.
4134 *
4135 *-----*
4136 * THIS PARAGRAPH WILL SELECT INFO FROM THE EXIT AND VEHL TABLES *
4137 * NEEDED TO PRINT ON PAGE 5 SCHEDULE OF AUTOMOBILES AND SCHEDULE *
4138 * OF PREM ON THE AUTO POLICY. *
4139 *-----*
4140 *
4141 2625-PROC-AUTO-INFO.
4142 *
4143 *
4144 IF WS-INFC-HSTP-GB-CODE-OF = 'NP'

```

```

4145         MOVE WS-HOLD-PREV-INFC-GB-NO TO WS-PPRT-INFC-GB-NO
4146     END-IF.
4147 *
4148     INITIALIZE
4149         WS-VHDS-AU-SERIAL-NO
4150         WS-VHDS-AU-MAKE
4151         WS-VHDS-AU-MODEL.
4152 *
4153     EXEC SQL AT ORA1
4154         SELECT VHDS_AU_SERIAL_NO,
4155                VHDS_AU_MAKE,
4156                VHDS_AU_MODEL
4157         INTO :WS-VHDS-AU-SERIAL-NO,
4158                :WS-VHDS-AU-MAKE,
4159                :WS-VHDS-AU-MODEL
4160         FROM VHDS
4161         WHERE VHDS_AU_NO = :WS-EXIT-VHDS-AU-NO
4162     END-EXEC.
4163 *
4164     IF SQLCODE NOT EQUAL 0
4165         MOVE "*** ERROR ** SELECTING VEHICLE_DESC TABLE INFO - 2625"
4166         TO WS-ERR-MSG
4167         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4168     END-IF.
4169 *
4170     INITIALIZE
4171         WS-VHCD-AU-YEAR
4172         WS-VHCD-AU-COST
4173         WS-VHCD-AU-USE
4174         WS-VHCD-AU-WEIGHT
4175         WS-VHCD-AU-CLASS-CODE
4176         WS-VHCD-AU-ZONE-BEGIN
4177         WS-VHCD-AU-ID-NO
4178         WS-VHCD-MCTB-AU-CLASS.
4179 *
4180     EXEC SQL AT ORA1
4181         SELECT VHCD_MCTB_AU_USE,
4182                MCTB_GB_EDIT,
4183                VHCD_AU_CLASS_CODE,
4184                VHCD_MCTB_AU_ZONE_BEGIN,
4185                VHCD_AU_YEAR,
4186                VHCD_AU_COST_NEW,
4187                VHCD_AU_NO,
4188                VHCD_AU_ID_NO,
4189                VHCD_MCTB_AU_CLASS
4190         INTO :WS-VHCD-AU-USE,
4191                :WS-VHCD-AU-WEIGHT,
4192                :WS-VHCD-AU-CLASS-CODE,
4193                :WS-VHCD-AU-ZONE-BEGIN,
4194                :WS-VHCD-AU-YEAR,
4195                :WS-VHCD-AU-COST,
4196                :WS-VHCD-AU-NO,
4197                :WS-VHCD-AU-ID-NO,
4198                :WS-VHCD-MCTB-AU-CLASS
4199         FROM MCTB, VHCD
4200         WHERE VHCD_AU_NO      = :WS-EXIT-VHCD-AU-NO

```

```

4201         AND MCTB_GB_CODE      = VHCD_MCTB_AU_WEIGHT
4202         AND MCTB_GB_CODE_TYPE = 'AUCARWGT'
4203     END-EXEC.
4204 *
4205     IF  SQLCODE NOT EQUAL 0
4206     MOVE "*** ERROR ** SELECTING VEHICLE_CODE TABLE INFO - 2625"
4207         TO WS-ERR-MSG
4208     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4209     END-IF.
4210 *
4211     INITIALIZE
4212         WS-MCTB-GB-CODE,
4213         WS-MCTB-GB-EDIT,
4214         WS-MILEAGE.
4215 *
4216     EXEC SQL AT ORA1
4217         SELECT MCTB_GB_CODE,
4218             MCTB_GB_EDIT,
4219             SUBSTR(MCTB_GB_DESC,1,7)
4220         INTO :WS-MCTB-GB-CODE,
4221             :WS-MCTB-GB-EDIT,
4222             :WS-MILEAGE
4223         FROM MCTB, VHFC
4224         WHERE VHFC_MCTB_AU_CODE IN ('FCRAD', 'FCMILE')
4225             AND VHFC_VHCD_AU_NO = :WS-EXIT-VHCD-AU-NO
4226             AND MCTB_GB_CODE      = VHFC_MCTB_AU_CODE
4227             AND MCTB_GB_EDIT      = VHFC_AU_FACTOR
4228             AND MCTB_GB_CODE_TYPE = 'AUFCDIT'
4229     END-EXEC.
4230 *
4231     IF  SQLCODE NOT EQUAL 0
4232     AND SQLCODE NOT EQUAL 1403
4233     MOVE "*** ERROR ** SELECTING VHFC TABLE MILEAGE INFO" TO
4234         WS-ERR-MSG
4235     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4236     END-IF.
4237 *
4238     IF WS-MCTB-GB-CODE EQUAL 'FCRAD'
4239     IF WS-MCTB-GB-EDIT EQUAL 'NA'
4240     MOVE " NA" TO WS-MILEAGE
4241     ELSE
4242     MOVE "APPLIES" TO WS-MILEAGE
4243     END-IF
4244     END-IF.
4245 *
4246     INITIALIZE
4247         WS-POP-DENSITY
4248         WS-VCLD-AU-ZONE-RATED.
4249 *
4250     EXEC SQL AT ORA1
4251         SELECT VCLD_AU_ZONE_RATED
4252         INTO :WS-VCLD-AU-ZONE-RATED
4253         FROM (SELECT VCLD_AU_ZONE_RATED
4254             FROM VCLD
4255             WHERE VCLD_AU_CLASS_CODE
4256                 = :WS-VHCD-AU-CLASS-CODE

```

```

4257         AND VCLD_STAT_AU_STATE
4258         = :WS-LOCT-STAT-GB-ALPHA-STATE-IN
4259         AND VCLD_AU_EFF_DATE
4260         <= TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')
4261         ORDER BY VCLD_AU_EFF_DATE DESC)
4262     WHERE ROWNUM = 1
4263 END-EXEC.
4264 *
4265 IF     SQLCODE NOT EQUAL 0
4266     AND SQLCODE NOT EQUAL 1403
4267     MOVE "*** ERROR ** SELECTING ZONE RATED INFO1" TO
4268         WS-ERR-MSG
4269     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4270 END-IF.
4271 *
4272 IF SQLCODE EQUAL 1403
4273     EXEC SQL AT ORA1
4274         SELECT VCLD_AU_ZONE_RATED
4275             INTO :WS-VCLD-AU-ZONE-RATED
4276             FROM (SELECT VCLD_AU_ZONE_RATED
4277                 FROM VCLD
4278                 WHERE VCLD_AU_CLASS_CODE
4279                     = :WS-VHCD-AU-CLASS-CODE
4280                 AND VCLD_STAT_AU_STATE
4281                     = 'XX'
4282                 AND VCLD_AU_EFF_DATE
4283                     <= TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')
4284                 ORDER BY VCLD_AU_EFF_DATE DESC)
4285             WHERE ROWNUM = 1
4286     END-EXEC
4287 *
4288     IF SQLCODE NOT EQUAL 0
4289     MOVE "*** ERROR ** SELECTING ZONE RATED INFO2"
4290         TO WS-ERR-MSG
4291     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4292 END-IF
4293 END-IF.
4294 *
4295 IF WS-VCLD-AU-ZONE-RATED EQUAL 'N' THEN
4296     MOVE WS-LOCT-GB-TERR-CD TO WS-POP-DENSITY
4297 ELSE
4298     MOVE 'ZONE' TO WS-POP-DENSITY
4299 END-IF.
4300 *
4301 INITIALIZE
4302     WS-DRIVER-RECORD.
4303 *
4304 EXEC SQL AT ORA1
4305     SELECT
4306         DECODE(RTRIM(LTRIM(SUBSTR(MCTB_GB_DESC,1,12))),
4307             'DUI/RD/FEL','UNFAVORABLE',SUBSTR(MCTB_GB_DESC,1,12))
4308     INTO :WS-DRIVER-RECORD
4309     FROM MCTB, VHFC
4310     WHERE VHFC_MCTB_AU_CODE = 'FCREC'
4311     AND VHFC_VHCD_AU_NO = :WS-EXIT-VHCD-AU-NO
4312     AND MCTB_GB_CODE = VHFC_MCTB_AU_CODE

```



```

4313         AND MCTB_GB_EDIT    = VHFC_AU_FACTOR
4314         AND MCTB_GB_CODE_TYPE = 'AUFCDIT'
4315     END-EXEC.
4316 *
4317     IF    SQLCODE NOT EQUAL 0
4318     AND SQLCODE NOT EQUAL 1403
4319     MOVE "*** ERROR ** SELECTING VHFC TABLE DRIVER RECORD INFO"
4320         TO WS-ERR-MSG
4321     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4322     END-IF.
4323 *
4324     INITIALIZE
4325         WS-GARAGE-CITY
4326         WS-GARAGE-STATE.
4327 *
4328     EXEC SQL AT ORA1
4329         SELECT ZIPC_GB_CITY,
4330                ZIPC_STAT_GB_STATE
4331         INTO :WS-GARAGE-CITY,
4332                :WS-GARAGE-STATE
4333         FROM ZIPC
4334         WHERE ZIPC_GB_NO = :WS-LOCT-ZIPC-GB-NO
4335     END-EXEC.
4336 *
4337     IF SQLCODE NOT EQUAL 0
4338     MOVE "*** ERROR ** SELECTING ZIPC TABLE INFO" TO
4339         WS-ERR-MSG
4340     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4341     END-IF.
4342 *
4343     MOVE WS-VHCD-AU-ZONE-BEGIN
4344         TO WS-TBL-TERR-CD(WS-ENTRY-COUNT).
4345 *
4346     MOVE WS-LOCT-GB-TERR-CD
4347     TO WS-TBL-TERR-CD-OLD(WS-ENTRY-COUNT).
4348     MOVE WS-VHCD-AU-YEAR
4349     TO WS-TBL-YEAR(WS-ENTRY-COUNT).
4350     MOVE WS-VHCD-AU-ID-NO
4351     TO WS-TBL-ID-NO(WS-ENTRY-COUNT).
4352     MOVE WS-EXIT-VHDS-AU-NO
4353     TO WS-TBL-VHDS-AU-NO(WS-ENTRY-COUNT)
4354     MOVE WS-VHDS-AU-SERIAL-NO
4355     TO WS-TBL-SERIAL-NO(WS-ENTRY-COUNT)
4356     MOVE WS-VHDS-AU-MAKE
4357     TO WS-TBL-MAKE(WS-ENTRY-COUNT).
4358     MOVE WS-VHDS-AU-MODEL
4359     TO WS-TBL-MODEL(WS-ENTRY-COUNT).
4360     MOVE WS-VHCD-AU-COST
4361     TO WS-TBL-COST(WS-ENTRY-COUNT).
4362     MOVE WS-DRIVER-RECORD
4363     TO WS-TBL-DRIVER-REC(WS-ENTRY-COUNT).
4364     MOVE WS-MILEAGE
4365     TO WS-TBL-MILEAGE(WS-ENTRY-COUNT).
4366     MOVE WS-VHCD-AU-USE
4367     TO WS-TBL-USE(WS-ENTRY-COUNT).
4368     MOVE WS-VHCD-AU-WEIGHT

```

```

4369 TO WS-TBL-WEIGHT(W$-ENTRY-COUNT).
4370 MOVE WS-POP-DENSITY
4371 TO WS-TBL-POP-DENSITY(W$-ENTRY-COUNT).
4372 MOVE WS-GARAGE-CITY
4373 TO WS-TBL-CITY(W$-ENTRY-COUNT).
4374 MOVE WS-GARAGE-STATE
4375 TO WS-TBL-STATE(W$-ENTRY-COUNT).
4376 MOVE WS-VHCD-AU-CLASS-CODE
4377 TO WS-TBL-CLASS-CODE(W$-ENTRY-COUNT).
4378 MOVE WS-VHCD-MCTB-AU-CLASS
4379 TO WS-TBL-CLASS(W$-ENTRY-COUNT).
4380 *
4381 IF WS-ENTRY-COUNT = 1000
4382 MOVE "*** ERROR ** WS-SCHEDULE-AUTO-TABLE SIZE EXCEEDED."
4383 TO WS-ERR-MSG
4384 PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4385 END-IF;
4386 *
4387
4388 IF WS-INFC-HSTP-GB-CODE-OF = 'NP'
4389 MOVE WS-HOLD-PREV-INFC-GB-NO TO WS-PPRT-INFC-GB-NO
4390 MOVE 0 TO WS-PREV-INFC-GB-NO
4391 PERFORM 3000-GET-AUTO-PREM THRU 3000-EXIT
4392 MOVE WS-HOLD-PPRT-INFC-GB-NO TO WS-PPRT-INFC-GB-NO
4393 MOVE WS-HOLD-PREV-INFC-GB-NO TO WS-PREV-INFC-GB-NO
4394 ELSE
4395 PERFORM 3000-GET-AUTO-PREM THRU 3000-EXIT
4396 END-IF.
4397 *
4398 MOVE 'N' TO WS-LOSS-PAYEE.
4399 PERFORM 3300-LOAD-LP-TBL THRU 3300-EXIT.
4400 *
4401 IF WS-LOSS-PAYEE = 'Y'
4402 MOVE 'YES' TO WS-TBL-LOSS-PAYEE(W$-ENTRY-COUNT)
4403 ELSE
4404 MOVE 'NO' TO WS-TBL-LOSS-PAYEE(W$-ENTRY-COUNT)
4405 END-IF.
4406 *
4407 ADD 1 TO WS-ENTRY-COUNT.
4408 *
4409 PERFORM 2650-FETCH-AUTO-INFO THRU 2650-EXIT.
4410 *
4411 2625-EXIT.
4412 EXIT.
4413 *
4414 *****
4415 * THIS PARAGRAPH CHECKS FOR THE EXISTENCE OF SSL COVERAGE ON A *
4416 * NEW YORK POLICY. *
4417 *****
4418 *
4419 2627-SSL-CHECK.
4420 *
4421 EXEC SQL AT ORA1
4422 SELECT CVOP_EDIT
4423 INTO WS-SSL-EXIST
4424 FROM (SELECT CVOP_GB_MCTB_EDIT CVOP_EDIT

```

```

4425         FROM CVOP
4426         WHERE CVOP_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
4427         AND CVOP_STAT_GB_STATE = 'NY'
4428         AND CVOP_GB_MCTB_CODE = 'FCSSL')
4429     WHERE ROWNUM = 1
4430 END-EXEC.
4431 *
4432     IF SQLCODE EQUAL 1403 THEN
4433         MOVE "N" TO WS-SSL-EXIST
4434     END-IF.
4435 *
4436     IF SQLCODE NOT EQUAL 1403 AND
4437         SQLCODE NOT EQUAL 0 THEN
4438         MOVE "*** ERROR ** CHECKING SSL COVERAGE - 2627"
4439         TO WS-ERR-MSG
4440         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4441     END-IF.
4442 *
4443     2627-EXIT.
4444     EXIT.
4445 *-----*
4446 * THIS PARAGRAPH WILL FETCH FROM THE AUTO_CURS. *
4447 *-----*
4448 *
4449     2650-FETCH-AUTO-INFO.
4450
4451 *
4452     INITIALIZE
4453         WS-EXIT-VHCD-AU-NO
4454         WS-EXIT-VHDS-AU-NO
4455         WS-LOCT-ZIPC-GB-NO
4456         WS-LOCT-STAT-GB-ALPHA-STATE-IN
4457         WS-LOCT-GB-NO
4458         WS-LOCT-GB-TERR-CD
4459         WS-VHCD-AU-ID-NO.
4460 *
4461     EXEC SQL
4462         FETCH AUTO_CURS
4463         INTO :WS-EXIT-VHCD-AU-NO,
4464             :WS-EXIT-VHDS-AU-NO,
4465             :WS-LOCT-ZIPC-GB-NO,
4466             :WS-LOCT-STAT-GB-ALPHA-STATE-IN,
4467             :WS-LOCT-GB-NO,
4468             :WS-LOCT-GB-TERR-CD,
4469             :WS-VHCD-AU-ID-NO
4470     END-EXEC.
4471 *
4472     IF SQLCODE NOT EQUAL 0
4473     AND SQLCODE NOT EQUAL 1403
4474         MOVE "*** ERROR ** FETCHING FROM AUTO_CURS CURSOR"
4475         TO WS-ERR-MSG
4476         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4477     END-IF.
4478 *
4479
4480     IF SQLCODE EQUAL 1403

```

```

4481         MOVE 'Y' TO WS-EOC-AUTO
4482     END-IF.
4483
4484 *****
4485
4486     IF WS-ALRDY-DN = 'N'
4487         EXEC SQL AT ORA1
4488             SELECT DISTINCT COUNT(LOCT_STAT_GB_ALPHA_STATE_IN)
4489             INTO WS-GRG-STS
4490             FROM LOCT,
4491                  VHCD,
4492                  EXIT,
4493                  COIT
4494             WHERE COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
4495                   AND EXIT_GB_NO = COIT_EXIT_GB_NO
4496                   AND VHCD_AU_NO = EXIT_VHCD_AU_NO
4497                   AND LOCT_GB_NO = COIT_LOCT_GB_NO
4498                   AND EXIT_VHDS_AU_NO IS NOT NULL
4499             AND EXIT_GB_NO NOT IN
4500             (SELECT COIT_EXIT_GB_NO
4501              FROM COIT
4502              WHERE COIT_INFC_GB_NO = :WS-PREV-INFC-GB-NO)
4503         END-EXEC
4504
4505         MOVE 'Y' to WS-ALRDY-DN
4506
4507         IF WS-GRG-STS EQUAL 1
4508             MOVE 'Y' to WS-SNGL-ST
4509         END-IF
4510     END-IF.
4511
4512
4513
4514 *****
4515
4516
4517
4518 *
4519 *CHECK TERRITORY HISTORY TABLE FOR TERRITORY CODE
4520 *
4521     EXEC SQL AT ORA1
4522         SELECT TECO_GB_TERR_CODE
4523         INTO :WS-TERR-CD
4524         FROM TECO, TRHS
4525         WHERE TECO_GB_NO = TRHS_TECO_GB_NO
4526               AND TECO_STAT_GB_STATE = :WS-LOCT-STAT-GB-ALPHA-STATE-IN
4527               AND TECO_ARTP_GB_TYPE = 'BAP'
4528               AND TRHS_LOCT_GB_NO = :WS-LOCT-GB-NO
4529               AND TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')
4530               >= TRHS_GB_EFF_DATE
4531               AND TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')
4532               < NVL(TRHS_GB_EXP_DATE,
4533                   TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')+1)
4534               AND TRHS_GB_EFF_DATE !=
4535                   NVL(TRHS_GB_EXP_DATE,
4536                       TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')+1)

```

```

4537     END-EXEC.
4538 *
4539     IF SQLCODE NOT EQUAL 0
4540     AND SQLCODE NOT EQUAL 1403
4541         MOVE "*** ERROR ** SELECTING TERRITORY CODE"
4542         TO WS-ERR-MSG
4543         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4544     END-IF.
4545 *
4546     IF SQLCODE EQUAL 0 THEN
4547         MOVE WS-TERR-CD TO WS-LOCT-GB-TERR-CD.
4548 *
4549     2650-EXIT.
4550     EXIT.
4551 *
4552     2675-GET-USAGE-TYPE.
4553 *
4554     EXEC SQL AT ORA1
4555         SELECT VCLD_AU_USAGE_TYPE
4556         INTO :WS-VCLD-AU-USAGE-TYPE
4557         FROM (SELECT VCLD_AU_USAGE_TYPE
4558             FROM VCLD
4559             WHERE VCLD_AU_CLASS_CODE
4560             = :WS-VHCD-AU-CLASS-CODE
4561             AND VCLD_STAT_AU_STATE
4562             = :WS-VCLD-STAT-AU-STATE
4563             AND VCLD_AU_EFF_DATE
4564             <= TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')
4565             ORDER BY VCLD_AU_EFF_DATE DESC)
4566         WHERE ROWNUM = 1
4567     END-EXEC.
4568 *
4569     IF SQLCODE NOT EQUAL 0
4570     AND SQLCODE NOT EQUAL 1403
4571         MOVE "*** ERROR ** ON SELECT FROM VCLD"
4572         TO WS-ERR-MSG
4573         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4574     END-IF.
4575 *
4576     2675-EXIT.
4577     EXIT.
4578 *
4579 *-----*
4580 * THIS PARAGRAPH WILL SELECT LOSS PAYEE INFORMATION NEEDED TO *
4581 * PRINT ON PAGE 6 OF THE AUTO POLICY. THIS PARAGRAPH WILL BE *
4582 * BE PERFORMED FOR EACH VEHICLE WITH A LOSS PAYEE. *
4583 *-----*
4584 *
4585     2700-GET-LOSS-PAYEE.
4586 *
4587         MOVE 'Y' TO WS-LOSS-PAYEE-LOCT.
4588         PERFORM 2800-GET-ACCT-INFO THRU 2800-EXIT.
4589 *
4590     2700-EXIT.
4591     EXIT.
4592 *

```

```

4593 *-----*
4594 * THIS PARAGRAPH WILL SELECT BSOR, LOCT, AND ZIPC INFORMATION *
4595 * THE ACCOUNT NUMBER AND ORTD CODE PASSED TO THIS PARAGRAPH. *
4596 *-----*
4597 *
4598 2800-GET-ACCT-INFO.
4599 *
4600 INITIALIZE
4601     WS-BSOR-GB-NAME,
4602     WS-BSOR-GB-ADNAME,
4603     WS-BSOR-BSTP-GB-TYPE-CODE-OF.
4604 *
4605
4606 EXEC SQL AT ORA1
4607     SELECT UPPER(BSOR_GB_NAME),
4608            UPPER(BSOR_GB_ADNAME),
4609            BSOR_BSTP_GB_TYPE_CODE_OF
4610     INTO :WS-BSOR-GB-NAME,
4611          :WS-BSOR-GB-ADNAME,
4612          :WS-BSOR-BSTP-GB-TYPE-CODE-OF
4613     FROM BSOR
4614     WHERE :WS-BSOR-GB-ACCOUNT = BSOR_GB_ACCOUNT
4615 END-EXEC.
4616 *
4617 IF SQLCODE NOT EQUAL 0
4618     MOVE "*** ERROR ** SELECTING BSOR INFO - 2200"
4619     TO WS-ERR-MSG
4620     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4621 END-IF.
4622 *
4623 MOVE 'X' TO WS-LOCT-GB-PB-IND.
4624 *
4625 PERFORM 2850-GET-LOCT-INFO THRU 2850-EXIT.
4626 *
4627 IF SQLCODE NOT EQUAL 0
4628     AND WS-LOSS-PAYEE-LOCT = 'N'
4629     DISPLAY WS-BSOR-GB-ACCOUNT
4630     MOVE "*** ERROR ** SELECTING LOCATION INFO" TO
4631     WS-ERR-MSG
4632     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4633 END-IF.
4634 *
4635 IF SQLCODE = 1403
4636     AND WS-LOSS-PAYEE-LOCT = 'Y'
4637     MOVE 'Z' TO WS-LOCT-GB-PB-IND
4638     PERFORM 2850-GET-LOCT-INFO THRU 2850-EXIT
4639     IF SQLCODE NOT EQUAL 0
4640     DISPLAY WS-BSOR-GB-ACCOUNT
4641     MOVE "*** ERROR ** SELECTING LOCATION INFO" TO
4642     WS-ERR-MSG
4643     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4644     END-IF
4645 END-IF.
4646 *
4647 2800-EXIT.
4648 EXIT.

```



```

4649 *
4650 2850-GET-LOCT-INFO.
4651 *
4652     INITIALIZE
4653         WS-LOCT-GB-STREET1
4654         WS-LOCT-GB-STREET2
4655         WS-LOCT-BSFM-GB-TYPE-CODE-OF
4656         WS-ZIPC-GB-CITY
4657         WS-ZIPC-GB-ZIP
4658         WS-ZIPC-STAT-GB-STATE.
4659 *
4660     EXEC SQL AT ORA1
4661         SELECT UPPER(LOCT_GB_STREET1),
4662                UPPER(LOCT_GB_STREET2),
4663                LOCT_BSFM_GB_TYPE_CODE_OF,
4664                UPPER(ZIPC_GB_CITY),
4665                ZIPC_GB_ZIP,
4666                ZIPC_STAT_GB_STATE
4667         INTO :WS-LOCT-GB-STREET1,
4668                :WS-LOCT-GB-STREET2,
4669                :WS-LOCT-BSFM-GB-TYPE-CODE-OF,
4670                :WS-ZIPC-GB-CITY,
4671                :WS-ZIPC-GB-ZIP,
4672                :WS-ZIPC-STAT-GB-STATE
4673         FROM LOCT, ZIPC
4674         WHERE LOCT_BSOR_GB_ACCOUNT = :WS-BSOR-GB-ACCOUNT
4675               AND LOCT_ZIPC_GB_NO = ZIPC_GB_NO
4676               AND NVL(LOCT_GB_PB_IND,'Z') = :WS-LOCT-GB-PB-IND
4677     END-EXEC.
4678
4679
4680
4681
4682
4683
4684 *
4685     IF      SQLCODE NOT EQUAL 0
4686     AND SQLCODE NOT EQUAL 1403
4687         DISPLAY WS-BSOR-GB-ACCOUNT
4688         MOVE "*** ERROR ** SELECTING LOCATION INFO" TO
4689             WS-ERR-MSG
4690         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4691     END-IF.
4692 *
4693 2850-EXIT.
4694 EXIT.
4695 *
4696 *-----*
4697 * THIS PARAGRAPH WILL SELECT INFORMATION FROM THE EXIT FOR ITEMS *
4698 * FOUR AND FIVE ON PAGE 3 ON THE AUTO POLICY. *
4699 * *
4700 *-----*
4701 *
4702 2900-GET-HIRED-NON-OWNED-INFO.
4703 *
4704 *EXCLUDED HAU1, HAUM, NOUI, AND NOUM FROM THE LIABILITY LIMIT SELECT

```

BECAUSE

*EVEN THOUGH THE PREMIUM IS INCLUDED IN THE LIABILITY PREMIUM THE LIMIT FOR
*THESE COVERAGE TYPES IS THE UNINSURED/UNDERINSURED MOTORIST LIMIT.

*

EXEC SQL AT ORA1

SELECT DISTINCT(LPAD(AUCV_AU_LIMIT,7,'0'))

INTO :WS-LIAB-LIMIT

FROM AUCV, EXIT

WHERE AUCV_AU_NO = EXIT_AUCV_AU_NO

AND AUCV_AU_PLAN_NO IN ('2','3')

AND AUCV_AU_TYPE NOT IN

*****-- ('HAUI','HAUM','NOUI','NOUM')

('HAUI','HAUM','NOUI','NOUM','NOMED','HAMED')

AND EXIT_GB_NO IN

(SELECT COIT_EXIT_GB_NO

FROM COIT

WHERE COIT_INFC_GB_NO

= :WS-PPRT-INFC-GB-NO)

END-EXEC.

IF SQLCODE NOT EQUAL 1403

IF SQLCODE = 0

IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99

DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"

PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT

ELSE

STRING "**ERROR* UNABLE TO PRINT ALL ",

" LIMITS FOR LIAB"

DELIMITED BY SIZE

INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)

ADD 1 TO WS-ENTRY-COUNT-6

ELSE

MOVE "*** ERROR ** SELECTING AUTO LIMITS - 2900"

TO WS-ERR-MSG

PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT

GO TO 2900-EXIT

END-IF

END-IF.

*

EXEC SQL AT ORA1

SELECT SUM(EXIT_GB_WRIT_PREM)

INTO :WS-SUM-WRIT-PREM-HIRED

FROM AUCV, EXIT

WHERE AUCV_AU_NO = EXIT_AUCV_AU_NO

AND AUCV_AU_PLAN_NO = '2'

AND EXIT_GB_NO IN

(SELECT COIT_EXIT_GB_NO

FROM COIT

WHERE COIT_INFC_GB_NO

= :WS-PPRT-INFC-GB-NO)

END-EXEC.

*

IF SQLCODE NOT EQUAL 0

AND SQLCODE NOT EQUAL 1403

MOVE "*** ERROR ** SELECTING SUM OF HIRED AUTO - 2900"

TO WS-ERR-MSG

```

4760         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4761         GO TO 2900-EXIT
4762     END-IF.
4763 *
4764     EXEC SQL AT ORA1
4765         SELECT SUM(EXIT_GB_WRIT_PREM)
4766             INTO :WS-SUM-WRIT-PREM-NON-OWNED
4767             FROM AUCV, EXIT
4768             WHERE AUCV_AU_NO = EXIT_AUCV_AU_NO
4769                 AND AUCV_AU_PLAN_NO IN ('3','7')
4770                 AND EXIT_GB_NO IN
4771                     (SELECT COIT_EXIT_GB_NO
4772                      FROM COIT
4773                      WHERE COIT_INFC_GB_NO
4774                        = :WS-PPRT-INFC-GB-NO)
4775     END-EXEC.
4776 *
4777     IF     SQLCODE NOT EQUAL 0
4778         AND SQLCODE NOT EQUAL 1403
4779         MOVE "*** ERROR ** SELECTING SUM OF NON-OWNED - 2900"
4780             TO WS-ERR-MSG
4781         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4782     END-IF.
4783 *
4784     2900-EXIT.
4785     EXIT.
4786 *
4787     3000-GET-AUTO-PREM.
4788 *
4789     EXEC SQL
4790         OPEN AUTO_PREM_CURS
4791     END-EXEC.
4792 *
4793     MOVE 'N' TO WS-EOC-AUTO-PREM.
4794
4795
4796     PERFORM 3475-FETCH-AUTO-PREM-CURS THRU 3475-EXIT.
4797 *
4798     PERFORM 3400-PROC-AUTO-WRIT-PREM THRU 3400-EXIT
4799         UNTIL WS-EOC-AUTO-PREM = 'Y'.
4800 *
4801     MOVE WS-PREV-INFC-GB-NO TO WS-3000-PREV-INFC-GB-NO.
4802     MOVE 0                 TO WS-PREV-INFC-GB-NO.
4803 *
4804     EXEC SQL
4805         OPEN AUTO_PREM_CURS
4806     END-EXEC.
4807 *
4808     MOVE 'N' TO WS-EOC-AUTO-PREM.
4809     PERFORM 3475-FETCH-AUTO-PREM-CURS THRU 3475-EXIT.
4810 *
4811     PERFORM 3450-PROC-AUTO-NET-PREM THRU 3450-EXIT
4812         UNTIL WS-EOC-AUTO-PREM = 'Y'.
4813 *
4814     MOVE WS-3000-PREV-INFC-GB-NO TO WS-PREV-INFC-GB-NO.
4815

```

```

4816      *
4817      3000-EXIT.
4818      EXIT.
4819      *
4820      3300-LOAD-LP-TBL.
4821      *
4822      MOVE 'N' TO WS-EOC-VINT-SW.
4823      *
4824      EXEC SQL
4825          OPEN VINT_CURS
4826      END-EXEC.
4827      *
4828      PERFORM 3375-FETCH-VINT-CURS THRU 3375-EXIT.
4829      *
4830      PERFORM 3350-PROC-VINT-CURS THRU 3350-EXIT
4831          UNTIL WS-EOC-VINT-SW = 'Y'.
4832      *
4833      3300-EXIT.
4834      EXIT.
4835      *
4836      3350-PROC-VINT-CURS.
4837      *
4838          IF WS-INFC-HSTP-GB-CODE-OF = 'NP'
4839              IF WS-VINT-GB-EFF-DATE NOT EQUAL WS-INFC-GB-EFF-DATE
4840                  AND WS-VINT-GB-EXP-DATE NOT EQUAL WS-INFC-GB-EFF-DATE
4841                      PERFORM 3375-FETCH-VINT-CURS THRU 3375-EXIT
4842                      GO TO 3350-EXIT
4843              END-IF
4844          END-IF.
4845      *
4846      MOVE WS-VINT-BSOR-GB-NO-ENTITY TO WS-BSOR-GB-ACCOUNT.
4847      MOVE 'Y' TO WS-LOSS-PAYEE.
4848      *
4849      PERFORM 2700-GET-LOSS-PAYEE THRU 2700-EXIT.
4850      *
4851      EVALUATE WS-VINT-ORTD-GB-TYPE
4852          WHEN 'LP'
4853              PERFORM 3351-LOAD-LP-TABLE THRU 3351-EXIT
4854          WHEN 'LH'
4855              PERFORM 3351-LOAD-LP-TABLE THRU 3351-EXIT
4856      END-EVALUATE.
4857      *
4858      PERFORM 3352-LOAD-SCH-TABLE THRU 3352-EXIT.
4859      *
4860      PERFORM 3375-FETCH-VINT-CURS THRU 3375-EXIT.
4861      *
4862      3350-EXIT.
4863      EXIT.
4864      *
4865      3351-LOAD-LP-TABLE.
4866      *
4867      MOVE WS-AU-LIMIT-SPLIT-HOLD TO
4868          WS-LP-LIMIT-SPLIT-HOLD(WS-ENTRY-COUNT-3)
4869      MOVE WS-AU-LIMIT-HOLD TO WS-LP-LIMIT-HOLD(WS-ENTRY-COUNT-3)
4870      IF WS-VINT-GB-EXP-DATE = WS-INFC-GB-EFF-DATE
4871          MOVE WS-VINT-GB-EXP-DATE

```

```

4872         TO WS-LP-EXP-DATE(WS-ENTRY-COUNT-3)
4873     ELSE
4874         MOVE SPACES TO WS-LP-EXP-DATE(WS-ENTRY-COUNT-3)
4875     END-IF.
4876     MOVE WS-VINT-BSOR-GB-NO-ENTITY
4877     TO WS-LP-ACCT(WS-ENTRY-COUNT-3).
4878     MOVE WS-BSOR-GB-NAME
4879     TO WS-LP-NAME(WS-ENTRY-COUNT-3).
4880     MOVE WS-ENTRY-COUNT
4881     TO WS-SV-ENTRY-COUNT(WS-ENTRY-COUNT-3).
4882     MOVE WS-BSOR-GB-ADNAME
4883     TO WS-LP-ADNAME(WS-ENTRY-COUNT-3).
4884     MOVE WS-LOCT-GB-STREET1
4885     TO WS-LP-STREET1(WS-ENTRY-COUNT-3).
4886     MOVE WS-LOCT-GB-STREET2
4887     TO WS-LP-STREET2(WS-ENTRY-COUNT-3).
4888     MOVE WS-ZIPC-GB-CITY      TO WS-CITY.
4889     MOVE WS-ZIPC-STAT-GB-STATE TO WS-STATE.
4890     MOVE WS-ZIPC-GB-ZIP      TO WS-ZIP.
4891 *
4892     IF WS-ZIP(6:4) = SPACES
4893         STRING WS-CITY DELIMITED BY " "
4894             " ",
4895             WS-STATE,
4896             " ",
4897             WS-ZIP(1:5)
4898             DELIMITED BY SIZE
4899         INTO WS-LP-CITY-ST-ZIP(WS-ENTRY-COUNT-3)
4900     ELSE
4901         STRING WS-CITY DELIMITED BY " "
4902             " ",
4903             WS-STATE,
4904             " ",
4905             WS-ZIP(1:5),
4906             "-",
4907             WS-ZIP(6:4)
4908             DELIMITED BY SIZE
4909         INTO WS-LP-CITY-ST-ZIP(WS-ENTRY-COUNT-3)
4910     END-IF.
4911 *
4912     IF WS-ENTRY-COUNT-3 = 1000
4913         MOVE "*** ERROR ** WS-LOSS-PAYEE-TABLE SIZE EXCEEDED."
4914         TO WS-ERR-MSG
4915         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4916     END-IF;
4917 *
4918     ADD 1 TO WS-ENTRY-COUNT-3.
4919 *
4920     3351-EXIT.
4921     EXIT.
4922 *
4923     3352-LOAD-SCH-TABLE.
4924 *
4925     MOVE WS-VINT-BSOR-GB-NO-ENTITY
4926     TO WS-SCH-LP-ACCT(WS-ENTRY-COUNT-4).
4927     MOVE WS-BSOR-GB-NAME

```

```

4928     TO WS-SCH-LP-NAME(WS-ENTRY-COUNT-4).
4929     MOVE WS-VINT-ORTD-GB-TYPE
4930     TO WS-SCH-LP-ORTD(WS-ENTRY-COUNT-4).
4931     MOVE WS-VHCD-AU-ID-NO
4932     TO WS-SCH-LP-CAR-NO(WS-ENTRY-COUNT-4).
4933     MOVE WS-BSOR-GB-ADNAME
4934     TO WS-SCH-LP-ADNAME(WS-ENTRY-COUNT-4).
4935     MOVE WS-LOCT-GB-STREET1
4936     TO WS-SCH-LP-STREET1(WS-ENTRY-COUNT-4).
4937     MOVE WS-LOCT-GB-STREET2
4938     TO WS-SCH-LP-STREET2(WS-ENTRY-COUNT-4).
4939     MOVE WS-GARAGE-STATE
4940     TO WS-SCH-GARAGE-STATE(WS-ENTRY-COUNT-4).
4941     MOVE WS-ZIPC-GB-CITY      TO WS-CITY.
4942     MOVE WS-ZIPC-STAT-GB-STATE TO WS-STATE.
4943     MOVE WS-ZIPC-GB-ZIP      TO WS-ZIP.
4944 *
4945     IF WS-ZIP(6:4) = SPACES
4946         STRING WS-CITY DELIMITED BY " "
4947             " ",
4948             WS-STATE,
4949             " ",
4950             WS-ZIP(1:5)
4951             DELIMITED BY SIZE
4952         INTO WS-SCH-LP-CITY-ST-ZIP(WS-ENTRY-COUNT-4)
4953     ELSE
4954         STRING WS-CITY DELIMITED BY " "
4955             " ",
4956             WS-STATE,
4957             " ",
4958             WS-ZIP(1:5),
4959             "- ",
4960             WS-ZIP(6:4)
4961             DELIMITED BY SIZE
4962         INTO WS-SCH-LP-CITY-ST-ZIP(WS-ENTRY-COUNT-4)
4963     END-IF.
4964 *
4965     IF WS-ENTRY-COUNT-4 = 1000
4966     MOVE "*** ERROR ** WS-SCH-LOSS-PAYEE-TABLE SIZE EXCEEDED."
4967     TO WS-ERR-MSG
4968     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4969     END-IF;
4970 *
4971     ADD 1 TO WS-ENTRY-COUNT-4.
4972 *
4973     3352-EXIT.
4974     EXIT.
4975 *
4976     3375-FETCH-VINT-CURS.
4977 *
4978     INITIALIZE
4979         WS-VINT-BSOR-GB-NO-ENTITY
4980         WS-VINT-ORTD-GB-TYPE
4981         WS-VINT-GB-EFF-DATE
4982         WS-VINT-GB-EXP-DATE.
4983 *

```



```

4984 EXEC SQL
4985     FETCH VINT_CURS
4986     INTO :WS-VINT-BSOR-GB-NO-ENTITY,
4987         :WS-VINT-ORTD-GB-TYPE,
4988         :WS-VINT-GB-EFF-DATE,
4989         :WS-VINT-GB-EXP-DATE
4990 END-EXEC.
4991 *
4992 IF     SQLCODE NOT EQUAL 0
4993     AND SQLCODE NOT EQUAL 1403
4994     MOVE "*** ERROR ** FETCHING VINT_CURS CURSOR"
4995     TO WS-ERR-MSG
4996     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
4997 END-IF.
4998 *
4999 IF SQLCODE EQUAL 1403
5000     MOVE "Y" TO WS-EOC-VINT-SW
5001 END-IF.
5002 *
5003 3375-EXIT.
5004 EXIT.
5005 *
5006 3400-PROC-AUTO-WRIT-PREM.
5007 *
5008 * THE FOLLOWING IF STATEMENTS IS USED TO KEEP TRACK OF EXPOSURES THAT
5009 * WERE ADDED AND/OR DELETED IN THE CURRENT INFORCE. THIS IS USED
5010 * TO PRINT A VEHICLE ADDED, VEHICLE DROPPED, OR VEHICLE AMENDED
5011 * DESCRIPTION ON THE 'RV' DEC PAGE.
5012 *
5013 IF WS-INFC-HSTP-GB-CODE-OF = 'RV'
5014     IF WS-EXIT-GB-EFF-DATE = WS-INFC-GB-EFF-DATE
5015         AND WS-EXIT-GB-EXP-DATE = WS-INFC-GB-EXP-DATE
5016         ADD 1
5017         TO WS-TBL-ADDED-EXIT(WS-ENTRY-COUNT)
5018     END-IF
5019     IF WS-EXIT-GB-EXP-DATE = WS-INFC-GB-EFF-DATE
5020         ADD 1
5021         TO WS-TBL-DROPPED-EXIT(WS-ENTRY-COUNT)
5022     END-IF
5023 END-IF.
5024 *
5025 EVALUATE WS-AUCV-AU-TYPE
5026     WHEN 'SLL'
5027         ADD WS-EXIT-WRIT-PREM
5028         TO WS-TBL-LIAB-WRIT(WS-ENTRY-COUNT)
5029         ADD WS-EXIT-WRIT-PREM
5030         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5031         MOVE WS-AUCV-AU-LIMIT
5032         TO WS-TBL-LIAB-LT(WS-ENTRY-COUNT)
5033     WHEN 'BUY'
5034         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-BUYBACK
5035         MOVE 'Y' TO WS-BUY-EXIST(WS-ENTRY-COUNT)
5036         MOVE 'Y' TO WS-VAR-BUY-EXIST
5037     WHEN 'PD'
5038         ADD WS-EXIT-WRIT-PREM
5039         TO WS-TBL-PD-WRIT(WS-ENTRY-COUNT)

```

```

5040      ADD WS-EXIT-WRIT-PREM
5041      TO WS-SUM-WRIT-PREM-PD
5042      ADD WS-EXIT-WRIT-PREM
5043      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5044      MOVE WS-AUCV-AU-LIMIT-REAL
5045      TO WS-TBL-PD-LT(W$-ENTRY-COUNT)
5046  WHEN 'BI'
5047      ADD WS-EXIT-WRIT-PREM
5048      TO WS-TBL-BI-WRIT(W$-ENTRY-COUNT)
5049      ADD WS-EXIT-WRIT-PREM
5050      TO WS-SUM-WRIT-PREM-BI
5051      ADD WS-EXIT-WRIT-PREM
5052      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5053      MOVE WS-AUCV-AU-LIMIT-REAL
5054      TO WS-TBL-BI-LT(W$-ENTRY-COUNT)
5055  WHEN 'PIP'
5056      IF WS-COVERAGE-LOCT = 'NY' THEN
5057          ADD WS-EXIT-WRIT-PREM
5058          TO WS-TBL-NY-PIP-WRIT(W$-ENTRY-COUNT)
5059          ADD WS-EXIT-WRIT-PREM
5060          TO WS-NY-WRIT-PREM-PIP
5061      ELSE
5062          IF WS-COVERAGE-LOCT = 'KY' THEN
5063              IF WS-EXIT-GB-DEDUCTIBLE > 0 THEN
5064                  MOVE WS-EXIT-GB-DEDUCTIBLE
5065                  TO WS-PIP-DEDUCTIBLE
5066              END-IF
5067          END-IF
5068          ADD WS-EXIT-WRIT-PREM
5069          TO WS-TBL-PIP-WRIT(W$-ENTRY-COUNT)
5070          ADD WS-EXIT-WRIT-PREM
5071          TO WS-SUM-WRIT-PREM-PIP
5072      END-IF
5073      ADD WS-EXIT-WRIT-PREM
5074      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5075  WHEN 'OBEL'
5076      ADD WS-EXIT-WRIT-PREM
5077      TO WS-TBL-NY-PIP-WRIT(W$-ENTRY-COUNT)
5078      ADD WS-EXIT-WRIT-PREM
5079      TO WS-SUM-WRIT-PREM-OBEL
5080      ADD WS-EXIT-WRIT-PREM
5081      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5082  WHEN 'ADPIP'
5083      ADD WS-EXIT-WRIT-PREM
5084      TO WS-TBL-ADDED-PIP-WRIT(W$-ENTRY-COUNT)
5085      ADD WS-EXIT-WRIT-PREM
5086      TO WS-SUM-WRIT-PREM-ADDED-PIP
5087      ADD WS-EXIT-WRIT-PREM
5088      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5089  *32513.6762 AAP ADDED TO SEPERATE OUT NY ADDED PIP
5090  WHEN 'NYPIP'
5091      ADD WS-EXIT-WRIT-PREM
5092      TO WS-TBL-ADDED-PIP-WRIT(W$-ENTRY-COUNT)
5093      ADD WS-EXIT-WRIT-PREM
5094      TO WS-SUM-WRIT-PREM-NY-ADDED-PIP
5095      ADD WS-EXIT-WRIT-PREM

```

```

5096         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5097 *****--
5098     WHEN 'DSPIP'
5099         ADD WS-EXIT-WRIT-PREM
5100         TO WS-TBL-PIP-WRIT(WS-ENTRY-COUNT)
5101         ADD WS-EXIT-WRIT-PREM
5102         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5103     WHEN 'DSMED'
5104         ADD WS-EXIT-WRIT-PREM
5105         TO WS-TBL-MED-PAY-WRIT(WS-ENTRY-COUNT)
5106         ADD WS-EXIT-WRIT-PREM
5107         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5108         MOVE WS-AUCV-AU-LIMIT
5109         TO WS-TBL-MED-PAY-LT(WS-ENTRY-COUNT)
5110 *****--
5111     WHEN 'PPI'
5112         ADD WS-EXIT-WRIT-PREM
5113         TO WS-TBL-PPI-WRIT(WS-ENTRY-COUNT)
5114         ADD WS-EXIT-WRIT-PREM
5115         TO WS-SUM-WRIT-PREM-PPI
5116         ADD WS-EXIT-WRIT-PREM
5117         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5118     WHEN 'MED'
5119         ADD WS-EXIT-WRIT-PREM
5120         TO WS-TBL-MED-PAY-WRIT(WS-ENTRY-COUNT)
5121         ADD WS-EXIT-WRIT-PREM
5122         TO WS-SUM-WRIT-PREM-MED-PAY
5123         ADD WS-EXIT-WRIT-PREM
5124         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5125         MOVE WS-AUCV-AU-LIMIT
5126         TO WS-TBL-MED-PAY-LT(WS-ENTRY-COUNT)
5127     WHEN 'MEDEXP'
5128         ADD WS-EXIT-WRIT-PREM
5129         TO WS-TBL-MEDEXP-WRIT(WS-ENTRY-COUNT)
5130         ADD WS-EXIT-WRIT-PREM
5131         TO WS-SUM-WRIT-PREM-MEDEXP
5132         ADD WS-EXIT-WRIT-PREM
5133         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5134         MOVE WS-AUCV-AU-LIMIT
5135         TO WS-TBL-MEDEXP-LT(WS-ENTRY-COUNT)
5136         ADD WS-EXIT-WRIT-PREM
5137         TO WS-TBL-MEDEXP(WS-ENTRY-COUNT)
5138         MOVE 'Y' TO WS-VAR-MEDEXP-EXIST
5139     WHEN 'ILB'
5140         ADD WS-EXIT-WRIT-PREM
5141         TO WS-TBL-MEDEXP-WRIT(WS-ENTRY-COUNT)
5142         ADD WS-EXIT-WRIT-PREM
5143         TO WS-SUM-WRIT-PREM-MEDEXP
5144         ADD WS-EXIT-WRIT-PREM
5145         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5146         MOVE WS-AUCV-AU-LIMIT
5147         TO WS-TBL-MEDEXP-LT(WS-ENTRY-COUNT)
5148         MOVE WS-EXIT-WRIT-PREM
5149         TO WS-TBL-ILB(WS-ENTRY-COUNT)
5150         MOVE 'Y' TO WS-ILB-EXIST(WS-ENTRY-COUNT)
5151     WHEN '047'

```

```

5152         ADD WS-EXIT-WRIT-PREM
5153         TO WS-SUM-WRIT-PREM-RENTAL
5154     WHEN '047CP'
5155         ADD WS-EXIT-WRIT-PREM
5156         TO WS-SUM-WRIT-PREM-RENTAL
5157     WHEN '047CL'
5158         ADD WS-EXIT-WRIT-PREM
5159         TO WS-SUM-WRIT-PREM-RENTAL
5160     WHEN 'UM'
5161         ADD WS-EXIT-WRIT-PREM
5162         TO WS-TBL-UM-WRIT(W$-ENTRY-COUNT)
5163         ADD WS-EXIT-WRIT-PREM
5164         TO WS-SUM-WRIT-PREM-UM
5165         ADD WS-EXIT-WRIT-PREM
5166         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5167         MOVE WS-AUCV-AU-LIMIT-REAL
5168         TO WS-TBL-UM-LT(W$-ENTRY-COUNT)
5169     *32513.6762 AAP ADDED STUM AND SUM SO THEY PRINT UNDER THE CORRECT
    HEADINGS
5170     WHEN 'STUM'
5171         ADD WS-EXIT-WRIT-PREM
5172         TO WS-TBL-UM-WRIT(W$-ENTRY-COUNT)
5173         ADD WS-EXIT-WRIT-PREM
5174         TO WS-SUM-WRIT-PREM-UM
5175         ADD WS-EXIT-WRIT-PREM
5176         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5177         MOVE WS-AUCV-AU-LIMIT-REAL
5178         TO WS-TBL-UM-LT(W$-ENTRY-COUNT)
5179         MOVE "Y" TO WS-STUM-EXIST
5180     WHEN 'SUM'
5181         ADD WS-EXIT-WRIT-PREM
5182         TO WS-TBL-SUM-WRIT(W$-ENTRY-COUNT)
5183         ADD WS-EXIT-WRIT-PREM
5184         TO WS-SUM-WRIT-PREM-SUM
5185         ADD WS-EXIT-WRIT-PREM
5186         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5187         MOVE WS-AUCV-AU-LIMIT-REAL
5188         TO WS-TBL-SUM-LT(W$-ENTRY-COUNT)
5189         MOVE "Y" TO WS-NY-SUM-EXIST
5190     WHEN 'UMPD'
5191         ADD WS-EXIT-WRIT-PREM
5192         TO WS-TBL-UM-WRIT(W$-ENTRY-COUNT)
5193         ADD WS-EXIT-WRIT-PREM
5194         TO WS-SUM-WRIT-PREM-UM
5195         ADD WS-EXIT-WRIT-PREM
5196         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5197         IF WS-WA-EXIST = "Y" THEN
5198             MOVE "Y" TO WS-WA-UIPD-EXIST
5199         END-IF
5200     WHEN 'UI'
5201         ADD WS-EXIT-WRIT-PREM
5202         TO WS-SUM-WRIT-PREM-UI
5203         ADD WS-EXIT-WRIT-PREM
5204         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5205         IF WS-INFC-HSTP-GB-CODE-OF = 'RV'
5206             ADD WS-EXIT-WRIT-PREM

```

```
5207         TO WS-TBL-UI-WRIT(WS-ENTRY-COUNT)
5208     ELSE
5209         ADD WS-EXIT-WRIT-PREM
5210         TO WS-TBL-UM-WRIT(WS-ENTRY-COUNT)
5211     END-IF
5212     MOVE WS-AUCV-AU-LIMIT-REAL
5213     TO WS-TBL-UI-LT(WS-ENTRY-COUNT)
5214 WHEN 'COMP'
5215     MOVE WS-AUCV-AU-LIMIT
5216     TO WS-TBL-STATD-AMT(WS-ENTRY-COUNT)
5217     ADD WS-EXIT-WRIT-PREM
5218     TO WS-TBL-COMP-WRIT(WS-ENTRY-COUNT)
5219     ADD WS-EXIT-WRIT-PREM
5220     TO WS-SUM-WRIT-PREM-COMP
5221     ADD WS-EXIT-WRIT-PREM
5222     TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5223 WHEN 'GLASSL'
5224     ADD WS-EXIT-WRIT-PREM
5225     TO WS-TBL-COMP-WRIT(WS-ENTRY-COUNT)
5226     ADD WS-EXIT-WRIT-PREM
5227     TO WS-SUM-WRIT-PREM-COMP
5228     ADD WS-EXIT-WRIT-PREM
5229     TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5230     MOVE 'Y' to WS-GLASS-EXIST-L
5231 WHEN 'GLASSM'
5232     ADD WS-EXIT-WRIT-PREM
5233     TO WS-TBL-COMP-WRIT(WS-ENTRY-COUNT)
5234     ADD WS-EXIT-WRIT-PREM
5235     TO WS-SUM-WRIT-PREM-COMP
5236     ADD WS-EXIT-WRIT-PREM
5237     TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5238     MOVE 'Y' to WS-GLASS-EXIST-M
5239 WHEN '064'
5240     MOVE WS-AUCV-AU-LIMIT
5241     TO WS-TBL-STATD-AMT(WS-ENTRY-COUNT)
5242     ADD WS-EXIT-WRIT-PREM
5243     TO WS-TBL-COMP-WRIT(WS-ENTRY-COUNT)
5244     ADD WS-EXIT-WRIT-PREM
5245     TO WS-SUM-WRIT-PREM-COMP
5246     ADD WS-EXIT-WRIT-PREM
5247     TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5248 WHEN '014'
5249     MOVE WS-AUCV-AU-LIMIT
5250     TO WS-TBL-STATD-AMT(WS-ENTRY-COUNT)
5251     ADD WS-EXIT-WRIT-PREM
5252     TO WS-TBL-COMP-WRIT(WS-ENTRY-COUNT)
5253     ADD WS-EXIT-WRIT-PREM
5254     TO WS-SUM-WRIT-PREM-COMP
5255     ADD WS-EXIT-WRIT-PREM
5256     TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5257 WHEN 'SF'
5258     MOVE WS-AUCV-AU-LIMIT
5259     TO WS-TBL-STATD-AMT(WS-ENTRY-COUNT)
5260     ADD WS-EXIT-WRIT-PREM
5261     TO WS-TBL-SPEC-LOSS-WRIT(WS-ENTRY-COUNT)
5262     ADD WS-EXIT-WRIT-PREM
```

```
5263         TO WS-SUM-WRIT-PREM-SPEC
5264     ADD WS-EXIT-WRIT-PREM
5265         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5266 WHEN 'SFT'
5267     MOVE WS-AUCV-AU-LIMIT
5268         TO WS-TBL-STAT$-AMT(W$-ENTRY-COUNT)
5269     ADD WS-EXIT-WRIT-PREM
5270         TO WS-TBL-SPEC-LOSS-WRIT(W$-ENTRY-COUNT)
5271     ADD WS-EXIT-WRIT-PREM
5272         TO WS-SUM-WRIT-PREM-SPEC
5273     ADD WS-EXIT-WRIT-PREM
5274         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5275 WHEN 'SFTW'
5276     MOVE WS-AUCV-AU-LIMIT
5277         TO WS-TBL-STAT$-AMT(W$-ENTRY-COUNT)
5278     ADD WS-EXIT-WRIT-PREM
5279         TO WS-TBL-SPEC-LOSS-WRIT(W$-ENTRY-COUNT)
5280     ADD WS-EXIT-WRIT-PREM
5281         TO WS-SUM-WRIT-PREM-SPEC
5282     ADD WS-EXIT-WRIT-PREM
5283         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5284 WHEN 'SLSP'
5285     MOVE WS-AUCV-AU-LIMIT
5286         TO WS-TBL-STAT$-AMT(W$-ENTRY-COUNT)
5287     ADD WS-EXIT-WRIT-PREM
5288         TO WS-TBL-SPEC-LOSS-WRIT(W$-ENTRY-COUNT)
5289     ADD WS-EXIT-WRIT-PREM
5290         TO WS-SUM-WRIT-PREM-SPEC
5291     ADD WS-EXIT-WRIT-PREM
5292         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5293 WHEN 'SP'
5294     MOVE WS-AUCV-AU-LIMIT
5295         TO WS-TBL-STAT$-AMT(W$-ENTRY-COUNT)
5296     ADD WS-EXIT-WRIT-PREM
5297         TO WS-TBL-SPEC-LOSS-WRIT(W$-ENTRY-COUNT)
5298     ADD WS-EXIT-WRIT-PREM
5299         TO WS-SUM-WRIT-PREM-SPEC
5300     ADD WS-EXIT-WRIT-PREM
5301         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5302 WHEN 'SPEC'
5303     MOVE WS-AUCV-AU-LIMIT
5304         TO WS-TBL-STAT$-AMT(W$-ENTRY-COUNT)
5305     ADD WS-EXIT-WRIT-PREM
5306         TO WS-TBL-SPEC-LOSS-WRIT(W$-ENTRY-COUNT)
5307     ADD WS-EXIT-WRIT-PREM
5308         TO WS-SUM-WRIT-PREM-SPEC
5309     ADD WS-EXIT-WRIT-PREM
5310         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5311 WHEN 'COLL'
5312     ADD WS-EXIT-WRIT-PREM
5313         TO WS-TBL-COLL-WRIT(W$-ENTRY-COUNT)
5314     ADD WS-EXIT-WRIT-PREM
5315         TO WS-SUM-WRIT-PREM-COLL
5316     ADD WS-EXIT-WRIT-PREM
5317         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5318 WHEN '047'
```



```
5319      ADD WS-EXIT-WRIT-PREM
5320      TO WS-RENT-PREM-WRIT(W$-ENTRY-COUNT)
5321      MOVE 'Y' to WS-RENTL-EXIST(W$-ENTRY-COUNT)
5322      display '047 writ ' WS-EXIT-WRIT-PREM
5323  WHEN '047CP'
5324      ADD WS-EXIT-WRIT-PREM
5325      TO WS-RENT-PREM-WRIT(W$-ENTRY-COUNT)
5326      MOVE 'Y' to WS-RENTL-EXIST(W$-ENTRY-COUNT)
5327      display '047CP writ ' WS-EXIT-WRIT-PREM
5328  WHEN '047CL'
5329      ADD WS-EXIT-WRIT-PREM
5330      TO WS-RENT-PREM-WRIT(W$-ENTRY-COUNT)
5331      MOVE 'Y' to WS-RENTL-EXIST(W$-ENTRY-COUNT)
5332      display '047CL writ ' WS-EXIT-WRIT-PREM
5333  WHEN 'WAIVE'
5334      MOVE 'W' TO WS-TBL-WAIVER(W$-ENTRY-COUNT)
5335      ADD WS-EXIT-WRIT-PREM
5336      TO WS-TBL-COLL-WRIT(W$-ENTRY-COUNT)
5337      ADD WS-EXIT-WRIT-PREM
5338      TO WS-SUM-WRIT-PREM-COLL
5339      ADD WS-EXIT-WRIT-PREM
5340      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5341  WHEN 'BROAD'
5342      MOVE 'B' TO WS-TBL-WAIVER(W$-ENTRY-COUNT)
5343      ADD WS-EXIT-WRIT-PREM
5344      TO WS-TBL-COLL-WRIT(W$-ENTRY-COUNT)
5345      ADD WS-EXIT-WRIT-PREM
5346      TO WS-SUM-WRIT-PREM-COLL
5347      ADD WS-EXIT-WRIT-PREM
5348      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5349  WHEN 'LIMIT'
5350      MOVE 'L' TO WS-TBL-WAIVER(W$-ENTRY-COUNT)
5351      ADD WS-EXIT-WRIT-PREM
5352      TO WS-TBL-COLL-WRIT(W$-ENTRY-COUNT)
5353      ADD WS-EXIT-WRIT-PREM
5354      TO WS-SUM-WRIT-PREM-COLL
5355      ADD WS-EXIT-WRIT-PREM
5356      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5357  WHEN 'LIMF'
5358      MOVE 'L' TO WS-TBL-WAIVER(W$-ENTRY-COUNT)
5359      ADD WS-EXIT-WRIT-PREM
5360      TO WS-TBL-COLL-WRIT(W$-ENTRY-COUNT)
5361      ADD WS-EXIT-WRIT-PREM
5362      TO WS-SUM-WRIT-PREM-COLL
5363      ADD WS-EXIT-WRIT-PREM
5364      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5365  WHEN '032'
5366      ADD WS-EXIT-WRIT-PREM
5367      TO WS-TBL-TOW-WRIT(W$-ENTRY-COUNT)
5368      ADD WS-EXIT-WRIT-PREM
5369      TO WS-SUM-WRIT-PREM-TOW
5370      ADD WS-EXIT-WRIT-PREM
5371      TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5372  WHEN '033'
5373      ADD WS-EXIT-WRIT-PREM
5374      TO WS-TBL-TOW-WRIT(W$-ENTRY-COUNT)
```

```
5375         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5376         ADD WS-EXIT-WRIT-PREM
5377         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5378     WHEN '035'
5379         ADD WS-EXIT-WRIT-PREM
5380         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5381         ADD WS-EXIT-WRIT-PREM
5382         TO WS-SUM-WRIT-PREM-TOW
5383         ADD WS-EXIT-WRIT-PREM
5384         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5385     WHEN '036'
5386         ADD WS-EXIT-WRIT-PREM
5387         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5388         ADD WS-EXIT-WRIT-PREM
5389         TO WS-SUM-WRIT-PREM-TOW
5390         ADD WS-EXIT-WRIT-PREM
5391         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5392     WHEN '061'
5393         ADD WS-EXIT-WRIT-PREM
5394         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5395         ADD WS-EXIT-WRIT-PREM
5396         TO WS-SUM-WRIT-PREM-TOW
5397         ADD WS-EXIT-WRIT-PREM
5398         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5399     WHEN '062'
5400         ADD WS-EXIT-WRIT-PREM
5401         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5402         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5403         ADD WS-EXIT-WRIT-PREM
5404         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5405     WHEN '063'
5406         ADD WS-EXIT-WRIT-PREM
5407         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5408         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5409         ADD WS-EXIT-WRIT-PREM
5410         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5411     WHEN '065'
5412         ADD WS-EXIT-WRIT-PREM
5413         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5414         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5415         ADD WS-EXIT-WRIT-PREM
5416         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5417     WHEN '066'
5418         ADD WS-EXIT-WRIT-PREM
5419         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5420         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5421         ADD WS-EXIT-WRIT-PREM
5422         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5423     WHEN '722'
5424         ADD WS-EXIT-WRIT-PREM
5425         TO WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
5426         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5427         ADD WS-EXIT-WRIT-PREM
5428         TO WS-TBL-TL-PREM(WS-ENTRY-COUNT)
5429     WHEN '723'
5430         ADD WS-EXIT-WRIT-PREM
```

```

5431         TO WS-TBL-TOW-WRIT(W$-ENTRY-COUNT)
5432         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5433         ADD WS-EXIT-WRIT-PREM
5434         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5435     WHEN 'TOW'
5436         ADD WS-EXIT-WRIT-PREM
5437         TO WS-TBL-TOW-WRIT(W$-ENTRY-COUNT)
5438         ADD WS-EXIT-WRIT-PREM TO WS-SUM-WRIT-PREM-TOW
5439         ADD WS-EXIT-WRIT-PREM
5440         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5441     WHEN 'APDTH'
5442         ADD WS-EXIT-WRIT-PREM
5443         TO WS-TBL-ADDED-PIP-WRIT(W$-ENTRY-COUNT)
5444         ADD WS-EXIT-WRIT-PREM
5445         TO WS-SUM-WRIT-PREM-NY-ADDED-PIP
5446         ADD WS-EXIT-WRIT-PREM
5447         TO WS-TBL-TL-PREM(W$-ENTRY-COUNT)
5448     END-EVALUATE.
5449 *
5450
5451     PERFORM 3475-FETCH-AUTO-PREM-CURS THRU 3475-EXIT.
5452 *
5453     3400-EXIT.
5454     EXIT.
5455 *
5456     3450-PROC-AUTO-NET-PREM.
5457 *
5458     INITIALIZE
5459         WS-VCLD-AU-USAGE-TYPE.
5460 *
5461     IF    WS-INFC-HSTP-GB-CODE-OF NOT EQUAL 'RV'
5462         MOVE WS-GARAGE-STATE TO WS-VCLD-STAT-AU-STATE
5463         PERFORM 2675-GET-USAGE-TYPE THRU 2675-EXIT
5464         IF SQLCODE = 1403
5465             MOVE 'XX' TO WS-VCLD-STAT-AU-STATE
5466             PERFORM 2675-GET-USAGE-TYPE THRU 2675-EXIT
5467         END-IF
5468     END-IF.
5469 *
5470     MOVE WS-VCLD-AU-USAGE-TYPE
5471     TO WS-TBL-USAGE-TYPE(W$-ENTRY-COUNT).
5472 *
5473     ADD 1 TO WS-TBL-TL-EXIT(W$-ENTRY-COUNT).
5474 *
5475
5476
5477     EVALUATE WS-AUCV-AU-TYPE
5478     WHEN 'SLL'
5479         ADD WS-EXIT-NET-PREM
5480         TO WS-TBL-LIAB-NET(W$-ENTRY-COUNT)
5481         MOVE 'Y' TO WS-SLL-EXIST(W$-ENTRY-COUNT)
5482         MOVE 'Y' TO WS-LIAB-EXIST(W$-ENTRY-COUNT)
5483         IF    WS-AUCV-AU-LIMIT NOT EQUAL WS-LIAB-LIMIT(1)
5484             AND WS-AUCV-AU-LIMIT NOT EQUAL WS-LIAB-LIMIT(2)
5485             IF WS-LIAB-LIMIT(1) EQUAL "0000000"
5486                 MOVE WS-AUCV-AU-LIMIT TO WS-LIAB-LIMIT(1)

```

```

5487         ELSE
5488             IF WS-LIAB-LIMIT(2) EQUAL "00000000"
5489                 MOVE WS-AUCV-AU-LIMIT TO WS-LIAB-LIMIT(2)
5490             ELSE
5491                 IF WS-ERR-MSG-TBL(W$-ENTRY-COUNT-6) > 99
5492                     DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5493                     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5494                 ELSE
5495                     STRING "**ERROR* UNABLE TO PRINT ALL SLL ",
5496                         " LIMITS FOR LIAB "
5497                     DELIMITED BY SIZE
5498                     INTO WS-ERR-MSG-TBL(W$-ENTRY-COUNT-6)
5499                     ADD 1 TO WS-ENTRY-COUNT-6
5500                 END-IF
5501             END-IF
5502         END-IF
5503     END-IF
5504     MOVE WS-AUCV-AU-LIMIT
5505     TO WS-TBL-LIAB-LT(W$-ENTRY-COUNT)
5506     WHEN 'PD'
5507         ADD WS-EXIT-NET-PREM
5508         TO WS-TBL-PD-NET(W$-ENTRY-COUNT)
5509         MOVE 'Y' TO WS-SLL-EXIST(W$-ENTRY-COUNT)
5510         MOVE 'Y' TO WS-PD-EXIST(W$-ENTRY-COUNT)
5511         IF WS-AUCV-AU-LIMIT NOT EQUAL WS-LIAB-LIMIT(1)
5512             AND WS-AUCV-AU-LIMIT NOT EQUAL WS-LIAB-LIMIT(2)
5513             IF WS-LIAB-LIMIT(1) EQUAL "00000000"
5514                 MOVE WS-AUCV-AU-LIMIT TO WS-LIAB-LIMIT(1)
5515             ELSE
5516                 IF WS-LIAB-LIMIT(2) EQUAL "00000000"
5517                     MOVE WS-AUCV-AU-LIMIT TO WS-LIAB-LIMIT(2)
5518                 ELSE
5519                     IF WS-ERR-MSG-TBL(W$-ENTRY-COUNT-6) > 99
5520                         DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5521                         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5522                     ELSE
5523                         STRING "**ERROR* UNABLE TO PRINT ALL PD",
5524                             " LIMITS FOR LIAB"
5525                         DELIMITED BY SIZE
5526                         INTO WS-ERR-MSG-TBL(W$-ENTRY-COUNT-6)
5527                         ADD 1 TO WS-ENTRY-COUNT-6
5528                     END-IF
5529                 END-IF
5530             END-IF
5531         END-IF
5532         MOVE WS-AUCV-AU-LIMIT
5533         TO WS-TBL-LIAB-LT(W$-ENTRY-COUNT)
5534         MOVE WS-AUCV-AU-LIMIT-REAL
5535         TO WS-TBL-PD-LT(W$-ENTRY-COUNT)
5536         IF WS-AUCV-AU-LIMIT-SPLIT = 'Y'
5537             MOVE WS-AUCV-AU-SPLIT-LIMIT TO WS-PD-LIMIT-HOLD
5538             MOVE WS-AUCV-AU-LIMIT-SPLIT TO WS-AUCV-AU-LIMIT-SPLIT-PD
5539         END-IF
5540     WHEN 'BI'
5541         ADD WS-EXIT-NET-PREM
5542         TO WS-TBL-BI-NET(W$-ENTRY-COUNT)

```

```

5543 MOVE 'Y' TO WS-SLL-EXIST(WS-ENTRY-COUNT)
5544 MOVE 'Y' TO WS-BI-EXIST(WS-ENTRY-COUNT)
5545 IF WS-AUCV-AU-LIMIT NOT EQUAL WS-LIAB-LIMIT(1)
5546 AND WS-AUCV-AU-LIMIT NOT EQUAL WS-LIAB-LIMIT(2)
5547 IF WS-LIAB-LIMIT(1) EQUAL "0000000"
5548 MOVE WS-AUCV-AU-LIMIT TO WS-LIAB-LIMIT(1)
5549 ELSE
5550 IF WS-LIAB-LIMIT(2) EQUAL "0000000"
5551 MOVE WS-AUCV-AU-LIMIT TO WS-LIAB-LIMIT(2)
5552 ELSE
5553 IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5554 DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5555 PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5556 ELSE
5557 STRING "**ERROR* UNABLE TO PRINT ALL BI",
5558 " LIMITS FOR LIAB"
5559 DELIMITED BY SIZE
5560 INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
5561 ADD 1 TO WS-ENTRY-COUNT-6
5562 END-IF
5563 END-IF
5564 END-IF
5565 END-IF
5566 MOVE WS-AUCV-AU-LIMIT
5567 TO WS-TBL-LIAB-LT(WS-ENTRY-COUNT)
5568 MOVE WS-AUCV-AU-LIMIT-REAL
5569 TO WS-TBL-BI-LT(WS-ENTRY-COUNT)
5570 MOVE WS-AUCV-AU-LIMIT TO WS-AU-LIMIT-HOLD
5571 MOVE WS-AUCV-AU-LIMIT-SPLIT TO WS-AU-LIMIT-SPLIT-HOLD
5572 IF WS-AUCV-AU-LIMIT-SPLIT = 'Y'
5573 MOVE WS-AUCV-AU-SPLIT-LIMIT TO WS-BI-LIMIT-HOLD
5574 MOVE WS-AUCV-AU-LIMIT-SPLIT TO WS-AUCV-AU-LIMIT-SPLIT-BI
5575 MOVE WS-AUCV-AU-SPLIT-LIMIT TO WS-AU-LIMIT-HOLD
5576 END-IF
5577 WHEN 'PIP'
5578 ADD WS-EXIT-NET-PREM TO WS-TBL-PIP-NET(WS-ENTRY-COUNT)
5579 IF WS-COVERAGE-LOCT = 'NY' THEN
5580 MOVE 'Y' TO WS-VAR-NY-PIP-EXIST
5581 MOVE 'Y' TO WS-NY-PIP-EXIST(WS-ENTRY-COUNT)
5582 IF WS-NY-PIP-BASIC-LIMIT(1) EQUAL "0000000"
5583 MOVE WS-AUCV-AU-LIMIT
5584 TO WS-NY-PIP-BASIC-LIMIT(1)
5585 ELSE
5586 IF WS-NY-PIP-BASIC-LIMIT(2) EQUAL "0000000"
5587 AND WS-AUCV-AU-LIMIT NOT EQUAL
5588 WS-NY-PIP-BASIC-LIMIT(1)
5589 MOVE WS-AUCV-AU-LIMIT
5590 TO WS-NY-PIP-BASIC-LIMIT(2)
5591 END-IF
5592 END-IF
5593 ELSE
5594 MOVE 'Y' TO WS-PIP-EXIST(WS-ENTRY-COUNT)
5595 MOVE 'Y' TO WS-VAR-PIP-EXIST
5596 END-IF
5597 *****--
5598 EVALUATE WS-AUCV-AU-SUB-TYPE

```

```

5599 WHEN 'BASIC'
5600     IF WS-PIP-BASIC-LIMIT(1) EQUAL "0000000"
5601         MOVE WS-AUCV-AU-LIMIT
5602             TO WS-PIP-BASIC-LIMIT(1)
5603     ELSE
5604         IF WS-PIP-BASIC-LIMIT(2) EQUAL "0000000"
5605             AND WS-AUCV-AU-LIMIT NOT EQUAL
5606                 WS-PIP-BASIC-LIMIT(1)
5607             MOVE WS-AUCV-AU-LIMIT
5608                 TO WS-PIP-BASIC-LIMIT(2)
5609     END-IF
5610 END-IF
5611 WHEN 'PDTH'
5612     IF WS-PIP-DEATH-LIMIT(1) EQUAL "0000000"
5613         MOVE WS-AUCV-AU-LIMIT
5614             TO WS-PIP-DEATH-LIMIT(1)
5615     ELSE
5616         IF WS-PIP-DEATH-LIMIT(2) EQUAL "0000000"
5617             AND WS-AUCV-AU-LIMIT NOT EQUAL
5618                 WS-PIP-DEATH-LIMIT(1)
5619             MOVE WS-AUCV-AU-LIMIT
5620                 TO WS-PIP-DEATH-LIMIT(2)
5621     END-IF
5622 END-IF
5623 WHEN 'POTH'
5624     IF WS-PIP-EXP-LIMIT(1) EQUAL "0000000"
5625         AND WS-AUCV-AU-LIMIT NOT EQUAL
5626             WS-PIP-EXP-LIMIT(1)
5627         MOVE WS-AUCV-AU-LIMIT
5628             TO WS-PIP-EXP-LIMIT(1)
5629     ELSE
5630         IF WS-PIP-EXP-LIMIT(2) EQUAL "0000000"
5631             MOVE WS-AUCV-AU-LIMIT
5632                 TO WS-PIP-EXP-LIMIT(2)
5633     END-IF
5634 END-IF
5635 WHEN 'PWRK'
5636     IF WS-PIP-WORK-LIMIT(1) EQUAL "0000000"
5637         AND WS-AUCV-AU-LIMIT NOT EQUAL
5638             WS-PIP-WORK-LIMIT(1)
5639         MOVE WS-AUCV-AU-LIMIT
5640             TO WS-PIP-WORK-LIMIT(1)
5641     ELSE
5642         IF WS-PIP-WORK-LIMIT(2) EQUAL "0000000"
5643             MOVE WS-AUCV-AU-LIMIT
5644                 TO WS-PIP-WORK-LIMIT(2)
5645     END-IF
5646 END-IF
5647 END-EVALUATE
5648 *****_
5649 WHEN 'DSPIP'
5650 ADD WS-EXIT-NET-PREM TO WS-TBL-PIP-NET(WS-ENTRY-COUNT)
5651 WHEN 'DSMED'
5652 ADD WS-EXIT-NET-PREM TO WS-TBL-MED-PAY-NET(WS-ENTRY-COUNT)
5653 WHEN 'ADPIP'
5654 ADD WS-EXIT-NET-PREM

```



```

5655         TO WS-TBL-ADDED-PIP-NET(WS-ENTRY-COUNT)
5656     MOVE 'Y' TO WS-ADPIP-EXIST(WS-ENTRY-COUNT)
5657     MOVE 'Y' TO WS-VAR-ADPIP-EXIST
5658     EVALUATE WS-AUCV-AU-SUB-TYPE
5659         WHEN 'APBAS'
5660             IF WS-APIP-BASIC-LIMIT(1) EQUAL "0000000"
5661                 MOVE WS-AUCV-AU-LIMIT
5662                 TO WS-APIP-BASIC-LIMIT(1)
5663             ELSE
5664                 IF WS-APIP-BASIC-LIMIT(2) EQUAL "0000000"
5665                     MOVE WS-AUCV-AU-LIMIT
5666                     TO WS-APIP-BASIC-LIMIT(2)
5667             END-IF
5668         END-IF
5669     END-EVALUATE
5670 *32513.6762 AAP ADDED TO PREVENT NY ADDED PIP FROM BEING LEFT OUT OF
TOTALS
5671     WHEN 'NYPIP'
5672         ADD WS-EXIT-NET-PREM
5673         TO WS-TBL-ADDED-PIP-NET(WS-ENTRY-COUNT)
5674     MOVE 'Y' TO WS-ADPIP-EXIST(WS-ENTRY-COUNT)
5675     MOVE 'Y' TO WS-VAR-ADPIP-EXIST
5676     MOVE 'Y' TO WS-NYAPIP-EXIST
5677     IF WS-NY-APIP-BASIC-LIMIT(1) EQUAL "0000000"
5678         IF WS-AUCV-AU-LIMIT NOT EQUAL SPACES THEN
5679             MOVE WS-AUCV-AU-LIMIT
5680             TO WS-NY-APIP-BASIC-LIMIT(1)
5681         END-IF
5682     ELSE
5683         IF WS-NY-APIP-BASIC-LIMIT(2) EQUAL "0000000"
5684             IF WS-AUCV-AU-LIMIT NOT EQUAL SPACES THEN
5685                 MOVE WS-AUCV-AU-LIMIT
5686                 TO WS-NY-APIP-BASIC-LIMIT(2)
5687             END-IF
5688         END-IF
5689     END-IF
5690     WHEN 'OBEL'
5691         ADD WS-EXIT-NET-PREM
5692         TO WS-TBL-OBEL-NET(WS-ENTRY-COUNT)
5693         ADD WS-EXIT-NET-PREM
5694         TO WS-TBL-PIP-NET(WS-ENTRY-COUNT)
5695         MOVE 'Y' TO WS-OBEL-EXIST(WS-ENTRY-COUNT)
5696         IF WS-OBEL-LIMIT(1) EQUAL "0000000"
5697             MOVE WS-AUCV-AU-LIMIT TO WS-OBEL-LIMIT(1)
5698         ELSE
5699             IF WS-OBEL-LIMIT(2) EQUAL "0000000"
5700                 MOVE WS-AUCV-AU-LIMIT TO WS-OBEL-LIMIT(2)
5701             END-IF
5702         END-IF
5703     WHEN 'PPI'
5704         ADD WS-EXIT-NET-PREM
5705         TO WS-TBL-PPI-NET(WS-ENTRY-COUNT)
5706         MOVE 'Y' TO WS-PPI-EXIST(WS-ENTRY-COUNT)
5707         MOVE 'Y' TO WS-VAR-PPI-EXIST
5708     WHEN 'MED'
5709         ADD WS-EXIT-NET-PREM

```

```

5710      TO WS-TBL-MED-PAY-NET(WS-ENTRY-COUNT)
5711      MOVE 'Y' TO WS-MED-EXIST(WS-ENTRY-COUNT)
5712      IF WS-AUCV-AU-LIMIT NOT EQUAL WS-MED-PAY-LIMIT(1)
5713      AND WS-AUCV-AU-LIMIT NOT EQUAL WS-MED-PAY-LIMIT(2)
5714          IF WS-MED-PAY-LIMIT(1) EQUAL "0000000"
5715              MOVE WS-AUCV-AU-LIMIT TO WS-MED-PAY-LIMIT(1)
5716          ELSE
5717              IF WS-MED-PAY-LIMIT(2) EQUAL "0000000"
5718                  MOVE WS-AUCV-AU-LIMIT
5719                  TO WS-MED-PAY-LIMIT(2)
5720              ELSE
5721                  IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5722                      DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5723                      PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5724                  ELSE
5725                      STRING "**ERROR* UNABLE TO PRINT ALL ",
5726                          " LIMITS FOR MED"
5727                      DELIMITED BY SIZE
5728                      INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
5729                      ADD 1 TO WS-ENTRY-COUNT-6
5730                  END-IF
5731              END-IF
5732          END-IF
5733      END-IF
5734      MOVE WS-AUCV-AU-LIMIT
5735      TO WS-TBL-MED-PAY-LT(WS-ENTRY-COUNT)
5736      WHEN 'MEDEXP'
5737          ADD WS-EXIT-NET-PREM
5738          TO WS-TBL-MEDEXP-NET(WS-ENTRY-COUNT)
5739          MOVE 'Y' TO WS-MEDEXP-EXIST(WS-ENTRY-COUNT)
5740          IF WS-AUCV-AU-LIMIT NOT EQUAL WS-MEDEXP-LIMIT(1)
5741          AND WS-AUCV-AU-LIMIT NOT EQUAL WS-MEDEXP-LIMIT(2)
5742              IF WS-MED-PAY-LIMIT(1) EQUAL "0000000"
5743                  MOVE WS-AUCV-AU-LIMIT TO WS-MEDEXP-LIMIT(1)
5744              ELSE
5745                  IF WS-MEDEXP-LIMIT(2) EQUAL "0000000"
5746                      MOVE WS-AUCV-AU-LIMIT
5747                      TO WS-MEDEXP-LIMIT(2)
5748                  ELSE
5749                      IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5750                          DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5751                          PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5752                      ELSE
5753                          STRING "**ERROR* UNABLE TO PRINT ALL ",
5754                              " LIMITS FOR MED"
5755                          DELIMITED BY SIZE
5756                          INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
5757                          ADD 1 TO WS-ENTRY-COUNT-6
5758                      END-IF
5759                  END-IF
5760              END-IF
5761          END-IF
5762          MOVE WS-AUCV-AU-LIMIT
5763          TO WS-TBL-MEDEXP-LT(WS-ENTRY-COUNT)
5764      WHEN 'UM'
5765          ADD WS-EXIT-NET-PREM TO WS-TBL-UM-NET(WS-ENTRY-COUNT)

```

```

5766 MOVE 'Y' TO WS-UM-EXIST(WS-ENTRY-COUNT)
5767 IF WS-AUCV-AU-LIMIT NOT EQUAL WS-UM-LIMIT(1)
5768 AND WS-AUCV-AU-LIMIT NOT EQUAL WS-UM-LIMIT(2)
5769 IF WS-UM-LIMIT(1) EQUAL "0000000"
5770 MOVE WS-AUCV-AU-LIMIT TO WS-UM-LIMIT(1)
5771 ELSE
5772 IF WS-UMPD-LIMIT1 = 'Y' THEN
5773 MOVE WS-UM-LIMIT(1) TO WS-UM-LIMIT(2)
5774 MOVE WS-AUCV-AU-LIMIT TO WS-UM-LIMIT(1)
5775 ELSE
5776 IF WS-UM-LIMIT(2) EQUAL "0000000"
5777 MOVE WS-AUCV-AU-LIMIT TO WS-UM-LIMIT(2)
5778 ELSE
5779 IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5780 DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5781 PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5782 ELSE
5783 STRING "**ERROR* UNABLE TO PRINT ALL ",
5784 " LIMITS FOR UM"
5785 DELIMITED BY SIZE
5786 INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
5787 ADD 1 TO WS-ENTRY-COUNT-6
5788 END-IF
5789 END-IF
5790 END-IF
5791 END-IF
5792 END-IF
5793 MOVE WS-AUCV-AU-LIMIT-REAL
5794 TO WS-TBL-UM-LT(WS-ENTRY-COUNT)
5795 IF WS-AUCV-AU-LIMIT-SPLIT = 'Y'
5796 MOVE WS-AUCV-AU-SPLIT-LIMIT TO WS-UM-LIMIT-HOLD
5797 MOVE WS-AUCV-AU-LIMIT-SPLIT
5798 TO WS-AUCV-AU-LIMIT-SPLIT-UM
5799 END-IF
5800 *32513.6762 AAP ADDED CODE TO PRINT NY STUM UNDER UNINSURED MOTORIST
COVERAGE
5801 WHEN 'STUM'
5802 ADD WS-EXIT-NET-PREM
5803 TO WS-TBL-UM-NET(WS-ENTRY-COUNT)
5804 MOVE 'Y' TO WS-UM-EXIST(WS-ENTRY-COUNT)
5805 IF WS-AUCV-AU-LIMIT NOT EQUAL WS-UM-LIMIT(1)
5806 AND WS-AUCV-AU-LIMIT NOT EQUAL WS-UM-LIMIT(2)
5807 IF WS-UM-LIMIT(1) EQUAL "0000000"
5808 MOVE WS-AUCV-AU-LIMIT TO WS-UM-LIMIT(1)
5809 ELSE
5810 IF WS-UM-LIMIT(2) EQUAL "0000000"
5811 MOVE WS-AUCV-AU-LIMIT TO WS-UM-LIMIT(2)
5812 ELSE
5813 IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5814 DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5815 PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5816 ELSE
5817 STRING "**ERROR* UNABLE TO PRINT ALL ",
5818 " LIMITS FOR UM"
5819 DELIMITED BY SIZE
5820 INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)

```

```

5821         ADD 1 TO WS-ENTRY-COUNT-6
5822     END-IF
5823 END-IF
5824 END-IF
5825 END-IF
5826 MOVE WS-AUCV-AU-LIMIT-REAL
5827     TO WS-TBL-UM-LT(WS-ENTRY-COUNT)
5828 IF WS-AUCV-AU-LIMIT-SPLIT = 'Y'
5829     MOVE WS-AUCV-AU-SPLIT-LIMIT TO WS-UM-LIMIT-HOLD
5830     MOVE WS-AUCV-AU-LIMIT-SPLIT
5831         TO WS-AUCV-AU-LIMIT-SPLIT-UM
5832 END-IF
5833 *32513.6762 AAP ADDED CODE TO PRINT NY SUM UNDER CORRECT HEADING
5834 WHEN 'SUM'
5835     ADD WS-EXIT-NET-PREM TO WS-TBL-SUM-NET(WS-ENTRY-COUNT)
5836     MOVE 'Y' TO WS-SUM-EXIST(WS-ENTRY-COUNT)
5837     IF WS-AUCV-AU-LIMIT NOT EQUAL WS-SUM-LIMIT(1)
5838         AND WS-AUCV-AU-LIMIT NOT EQUAL WS-SUM-LIMIT(2)
5839         IF WS-SUM-LIMIT(1) EQUAL "0000000"
5840             MOVE WS-AUCV-AU-LIMIT TO WS-SUM-LIMIT(1)
5841         ELSE
5842             IF WS-SUM-LIMIT(2) EQUAL "0000000"
5843                 MOVE WS-AUCV-AU-LIMIT TO WS-SUM-LIMIT(2)
5844             ELSE
5845                 IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5846                     DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5847                     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5848                 ELSE
5849                     STRING "**ERROR* UNABLE TO PRINT ALL ",
5850                         " LIMITS FOR SUM"
5851                     DELIMITED BY SIZE
5852                     INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
5853                     ADD 1 TO WS-ENTRY-COUNT-6
5854                 END-IF
5855             END-IF
5856         END-IF
5857     END-IF
5858     MOVE WS-AUCV-AU-LIMIT-REAL
5859         TO WS-TBL-SUM-LT(WS-ENTRY-COUNT)
5860     IF WS-AUCV-AU-LIMIT-SPLIT = 'Y'
5861         MOVE WS-AUCV-AU-SPLIT-LIMIT TO WS-SUM-LIMIT-HOLD
5862         MOVE WS-AUCV-AU-LIMIT-SPLIT
5863             TO WS-AUCV-AU-LIMIT-SPLIT-SUM
5864     END-IF
5865 WHEN 'UMPD'
5866     ADD WS-EXIT-NET-PREM
5867         TO WS-TBL-UM-NET(WS-ENTRY-COUNT)
5868     MOVE 'Y' TO WS-UM-EXIST(WS-ENTRY-COUNT)
5869     IF WS-AUCV-AU-LIMIT NOT EQUAL WS-UM-LIMIT(1)
5870         AND WS-AUCV-AU-LIMIT NOT EQUAL WS-UM-LIMIT(2)
5871         IF WS-UM-LIMIT(1) EQUAL "0000000"
5872             MOVE 'Y' TO WS-UMPD-LIMIT1
5873             MOVE WS-AUCV-AU-LIMIT TO WS-UM-LIMIT(1)
5874         ELSE
5875             IF WS-UM-LIMIT(2) EQUAL "0000000"
5876                 MOVE WS-AUCV-AU-LIMIT TO WS-UM-LIMIT(2)

```

```

5877     ELSE
5878     IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5879         DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5880         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5881     ELSE
5882         STRING "**ERROR* UNABLE TO PRINT ALL ",
5883             " LIMITS FOR UM"
5884         DELIMITED BY SIZE
5885         INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
5886         ADD 1 TO WS-ENTRY-COUNT-6
5887     END-IF
5888 END-IF
5889 END-IF
5890 END-IF
5891 WHEN 'UI'
5892     IF WS-INFC-HSTP-GB-CODE-OF = 'RV'
5893         ADD WS-EXIT-NET-PREM
5894         TO WS-TBL-UI-NET(WS-ENTRY-COUNT)
5895         MOVE 'Y' TO WS-UI-EXIST(WS-ENTRY-COUNT)
5896     ELSE
5897         ADD WS-EXIT-NET-PREM
5898         TO WS-TBL-UM-NET(WS-ENTRY-COUNT)
5899         MOVE 'Y' TO WS-UM-EXIST(WS-ENTRY-COUNT)
5900     END-IF
5901     IF WS-AUCV-AU-LIMIT NOT EQUAL WS-UI-LIMIT(1)
5902     AND WS-AUCV-AU-LIMIT NOT EQUAL WS-UI-LIMIT(2)
5903         IF WS-UI-LIMIT(1) EQUAL "0000000"
5904             MOVE WS-AUCV-AU-LIMIT TO WS-UI-LIMIT(1)
5905         ELSE
5906             IF WS-UI-LIMIT(2) EQUAL "0000000"
5907                 MOVE WS-AUCV-AU-LIMIT TO WS-UI-LIMIT(2)
5908             ELSE
5909                 IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
5910                     DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
5911                     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
5912                 ELSE
5913                     STRING "**ERROR* UNABLE TO PRINT ALL ",
5914                         " LIMITS FOR UI"
5915                     DELIMITED BY SIZE
5916                     INTO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
5917                     ADD 1 TO WS-ENTRY-COUNT-6
5918                 END-IF
5919             END-IF
5920         END-IF
5921     END-IF
5922     MOVE WS-AUCV-AU-LIMIT-REAL
5923     TO WS-TBL-UI-LT(WS-ENTRY-COUNT)
5924     IF WS-AUCV-AU-LIMIT-SPLIT = 'Y'
5925         MOVE WS-AUCV-AU-SPLIT-LIMIT TO WS-UI-LIMIT-HOLD
5926         MOVE WS-AUCV-AU-LIMIT-SPLIT TO WS-AUCV-AU-LIMIT-SPLIT-UI
5927     END-IF
5928 WHEN 'COMP'
5929     MOVE WS-EXIT-GB-DEDUCTIBLE
5930     TO WS-TBL-COMP-DEDUCT(WS-ENTRY-COUNT)
5931     ADD WS-EXIT-NET-PREM
5932     TO WS-TBL-COMP-NET(WS-ENTRY-COUNT)

```



```
5933         MOVE 'Y' TO WS-COMP-EXIST(WS-ENTRY-COUNT)
5934         MOVE 'Y' TO WS-VAR-COMP-EXIST
5935         MOVE WS-AUCV-AU-LIMIT
5936             TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5937     WHEN 'GLASSM'
5938         MOVE WS-EXIT-GB-DEDUCTIBLE
5939             TO WS-TBL-COMP-DEDUCT(WS-ENTRY-COUNT)
5940         ADD WS-EXIT-NET-PREM
5941             TO WS-TBL-COMP-NET(WS-ENTRY-COUNT)
5942         MOVE 'Y' TO WS-COMP-EXIST(WS-ENTRY-COUNT)
5943         MOVE 'Y' TO WS-VAR-COMP-EXIST
5944         MOVE 'Y' TO WS-GLASS-EXIST-M
5945     WHEN 'GLASSL'
5946         MOVE WS-EXIT-GB-DEDUCTIBLE
5947             TO WS-TBL-COMP-DEDUCT(WS-ENTRY-COUNT)
5948         ADD WS-EXIT-NET-PREM
5949             TO WS-TBL-COMP-NET(WS-ENTRY-COUNT)
5950         MOVE 'Y' TO WS-COMP-EXIST(WS-ENTRY-COUNT)
5951         MOVE 'Y' TO WS-VAR-COMP-EXIST
5952         MOVE 'Y' TO WS-GLASS-EXIST-L
5953     WHEN '064'
5954         MOVE WS-AUCV-AU-LIMIT
5955             TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5956         ADD WS-EXIT-NET-PREM
5957             TO WS-TBL-COMP-NET(WS-ENTRY-COUNT)
5958         MOVE 'Y' TO WS-COMP-EXIST(WS-ENTRY-COUNT)
5959         MOVE 'Y' TO WS-VAR-COMP-EXIST
5960     WHEN '014'
5961         MOVE WS-AUCV-AU-LIMIT
5962             TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5963         ADD WS-EXIT-NET-PREM
5964             TO WS-TBL-COMP-NET(WS-ENTRY-COUNT)
5965         MOVE 'Y' TO WS-COMP-EXIST(WS-ENTRY-COUNT)
5966         MOVE 'Y' TO WS-VAR-COMP-EXIST
5967     WHEN 'SF'
5968         MOVE WS-AUCV-AU-LIMIT
5969             TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5970         ADD WS-EXIT-NET-PREM
5971             TO WS-TBL-SPEC-LOSS-NET(WS-ENTRY-COUNT)
5972         MOVE 'Y' TO WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT)
5973         MOVE 'Y' TO WS-VAR-SPEC-LOSS-EXIST
5974     WHEN 'SFT'
5975         MOVE WS-AUCV-AU-LIMIT
5976             TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5977         ADD WS-EXIT-NET-PREM
5978             TO WS-TBL-SPEC-LOSS-NET(WS-ENTRY-COUNT)
5979         MOVE 'Y' TO WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT)
5980         MOVE 'Y' TO WS-VAR-SPEC-LOSS-EXIST
5981     WHEN 'SFTW'
5982         MOVE WS-AUCV-AU-LIMIT
5983             TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5984         ADD WS-EXIT-NET-PREM
5985             TO WS-TBL-SPEC-LOSS-NET(WS-ENTRY-COUNT)
5986         MOVE 'Y' TO WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT)
5987         MOVE 'Y' TO WS-VAR-SPEC-LOSS-EXIST
5988     WHEN 'SLSP'
```



```
5989      MOVE WS-AUCV-AU-LIMIT
5990      TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5991      ADD  WS-EXIT-NET-PREM
5992      TO WS-TBL-SPEC-LOSS-NET(WS-ENTRY-COUNT)
5993      MOVE 'Y' TO WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT)
5994      MOVE 'Y' TO WS-VAR-SPEC-LOSS-EXIST
5995  WHEN 'SP'
5996      MOVE WS-AUCV-AU-LIMIT
5997      TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
5998      ADD  WS-EXIT-NET-PREM
5999      TO WS-TBL-SPEC-LOSS-NET(WS-ENTRY-COUNT)
6000      MOVE 'Y' TO WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT)
6001      MOVE 'Y' TO WS-VAR-SPEC-LOSS-EXIST
6002  WHEN 'SPEC'
6003      MOVE WS-AUCV-AU-LIMIT
6004      TO WS-TBL-STATED-AMT(WS-ENTRY-COUNT)
6005      ADD  WS-EXIT-NET-PREM
6006      TO WS-TBL-SPEC-LOSS-NET(WS-ENTRY-COUNT)
6007      MOVE 'Y' TO WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT)
6008      MOVE 'Y' TO WS-VAR-SPEC-LOSS-EXIST
6009  WHEN 'COLL'
6010      MOVE WS-EXIT-GB-DEDUCTIBLE
6011      TO WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
6012      ADD  WS-EXIT-NET-PREM
6013      TO WS-TBL-COLL-NET(WS-ENTRY-COUNT)
6014      MOVE 'Y' TO WS-COLL-EXIST(WS-ENTRY-COUNT)
6015      MOVE 'N' TO WS-LIMF(WS-ENTRY-COUNT)
6016      MOVE 'Y' TO WS-VAR-COLL-EXIST
6017  WHEN 'WAIVE'
6018      ADD  WS-EXIT-NET-PREM
6019      TO WS-TBL-COLL-NET(WS-ENTRY-COUNT)
6020      MOVE 'Y' TO WS-COLL-EXIST(WS-ENTRY-COUNT)
6021      MOVE 'N' TO WS-LIMF(WS-ENTRY-COUNT)
6022      MOVE 'Y' TO WS-VAR-COLL-EXIST
6023      MOVE 'Y' TO WS-WAIVE-EXIST(WS-ENTRY-COUNT)
6024  WHEN 'BROAD'
6025      MOVE WS-EXIT-GB-DEDUCTIBLE
6026      TO WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
6027      ADD  WS-EXIT-NET-PREM
6028      TO WS-TBL-COLL-NET(WS-ENTRY-COUNT)
6029      MOVE 'Y' TO WS-COLL-EXIST(WS-ENTRY-COUNT)
6030      MOVE 'N' TO WS-LIMF(WS-ENTRY-COUNT)
6031      MOVE 'Y' TO WS-VAR-COLL-EXIST
6032  WHEN 'LIMIT'
6033      MOVE WS-EXIT-GB-DEDUCTIBLE
6034      TO WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
6035      ADD  WS-EXIT-NET-PREM
6036      TO WS-TBL-COLL-NET(WS-ENTRY-COUNT)
6037      MOVE 'Y' TO WS-COLL-EXIST(WS-ENTRY-COUNT)
6038      MOVE 'N' TO WS-LIMF(WS-ENTRY-COUNT)
6039      MOVE 'Y' TO WS-VAR-COLL-EXIST
6040  WHEN 'LIMF'
6041      MOVE WS-EXIT-GB-DEDUCTIBLE
6042      TO WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
6043      ADD  WS-EXIT-NET-PREM
6044      TO WS-TBL-COLL-NET(WS-ENTRY-COUNT)
```

```
6045         MOVE 'Y' TO WS-COLL-EXIST(WS-ENTRY-COUNT)
6046         MOVE 'Y' TO WS-LIMF(WS-ENTRY-COUNT)
6047         MOVE 'Y' TO WS-VAR-COLL-EXIST
6048     WHEN '047'
6049         ADD WS-EXIT-NET-PREM
6050         TO WS-RENT-PREM-NET(WS-ENTRY-COUNT)
6051         MOVE 'Y' TO WS-RENTL-EXIST(WS-ENTRY-COUNT)
6052         display '047 NET' WS-EXIT-NET-PREM
6053     WHEN '047CP'
6054         ADD WS-EXIT-NET-PREM
6055         TO WS-RENT-PREM-NET(WS-ENTRY-COUNT)
6056         MOVE 'Y' TO WS-RENTL-EXIST(WS-ENTRY-COUNT)
6057         display '047CP NET' WS-EXIT-NET-PREM
6058     WHEN '047CL'
6059         ADD WS-EXIT-NET-PREM
6060         TO WS-RENT-PREM-NET(WS-ENTRY-COUNT)
6061         MOVE 'Y' TO WS-RENTL-EXIST(WS-ENTRY-COUNT)
6062         display '047CL NET' WS-EXIT-NET-PREM
6063     WHEN '032'
6064         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6065         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6066         MOVE 'Y' TO WS-VAR-TOW-EXIST
6067     WHEN '033'
6068         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6069         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6070         MOVE 'Y' TO WS-VAR-TOW-EXIST
6071     WHEN '035'
6072         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6073         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6074         MOVE 'Y' TO WS-VAR-TOW-EXIST
6075     WHEN '036'
6076         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6077         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6078         MOVE 'Y' TO WS-VAR-TOW-EXIST
6079     WHEN '061'
6080         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6081         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6082         MOVE 'Y' TO WS-VAR-TOW-EXIST
6083     WHEN '062'
6084         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6085         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6086         MOVE 'Y' TO WS-VAR-TOW-EXIST
6087     WHEN '063'
6088         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6089         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6090         MOVE 'Y' TO WS-VAR-TOW-EXIST
6091     WHEN '065'
6092         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6093         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6094         MOVE 'Y' TO WS-VAR-TOW-EXIST
6095     WHEN '066'
6096         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6097         MOVE 'Y' TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6098         MOVE 'Y' TO WS-VAR-TOW-EXIST
6099     WHEN '722'
6100         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
```

```

6101         MOVE 'Y'      TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6102         MOVE 'Y'      TO WS-VAR-TOW-EXIST
6103     WHEN '723'
6104         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6105         MOVE 'Y'      TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6106         MOVE 'Y'      TO WS-VAR-TOW-EXIST
6107     WHEN 'TOW'
6108         ADD WS-EXIT-NET-PREM TO WS-TBL-TOW-NET(WS-ENTRY-COUNT)
6109         MOVE 'Y'      TO WS-TOW-EXIST(WS-ENTRY-COUNT)
6110         MOVE 'Y'      TO WS-VAR-TOW-EXIST
6111     WHEN 'APDTH'
6112         ADD WS-EXIT-NET-PREM
6113         TO WS-TBL-ADDED-PIP-NET(WS-ENTRY-COUNT)
6114         MOVE 'Y' TO WS-ADPIP-EXIST(WS-ENTRY-COUNT)
6115         MOVE 'Y' TO WS-VAR-ADPIP-EXIST
6116         MOVE 'Y' TO WS-NYAPDTH-EXIST
6117         IF WS-NY-APDTH-ADD-LIMIT(1) EQUAL "0000000"
6118             IF WS-AUCV-AU-LIMIT NOT EQUAL SPACES THEN
6119                 MOVE WS-AUCV-AU-LIMIT
6120                 TO WS-NY-APDTH-ADD-LIMIT(1)
6121             END-IF
6122         END-IF
6123     END-EVALUATE.
6124 *
6125     PERFORM 3475-FETCH-AUTO-PREM-CURS THRU 3475-EXIT.
6126 *
6127     3450-EXIT.
6128     EXIT.
6129 *
6130     3475-FETCH-AUTO-PREM-CURS.
6131
6132 *
6133     INITIALIZE
6134         WS-EXIT-GB-NO,
6135         WS-EXIT-NET-PREM,
6136         WS-EXIT-WRIT-PREM,
6137         WS-EXIT-GB-DEDUCTIBLE,
6138         WS-AUCV-AU-TYPE,
6139         WS-AUCV-AU-SUB-TYPE,
6140         WS-AUCV-AU-LIMIT,
6141         WS-AUCV-AU-LIMIT-REAL,
6142         WS-AUCV-AU-LIMIT-SPLIT,
6143         WS-EXIT-AUBP-AU-NO,
6144         WS-EXIT-GB-EFF-DATE,
6145         WS-EXIT-GB-EXP-DATE,
6146         WS-COVERAGE-LOCT.
6147 *
6148     EXEC SQL
6149         FETCH AUTO_PREM_CURS
6150         INTO :WS-EXIT-GB-NO,
6151             :WS-EXIT-NET-PREM,
6152             :WS-EXIT-WRIT-PREM,
6153             :WS-EXIT-GB-DEDUCTIBLE,
6154             :WS-AUCV-AU-TYPE,
6155             :WS-AUCV-AU-SUB-TYPE,
6156             :WS-AUCV-AU-LIMIT,

```

```

6157             :WS-AUCV-AU-LIMIT-REAL,
6158             :WS-AUCV-AU-LIMIT-SPLIT,
6159             :WS-EXIT-AUBP-AU-NO,
6160             :WS-EXIT-GB-EFF-DATE,
6161             :WS-EXIT-GB-EXP-DATE,
6162             :WS-AUCV-AU-SPLIT-LIMIT,
6163             :WS-COVERAGE-LOCT
6164 END-EXEC.
6165
6166 *
6167 IF     SQLCODE NOT EQUAL 0
6168     AND SQLCODE NOT EQUAL 1403
6169     MOVE "*** ERROR ** FETCHING AUTO_PREM_CURS CURSOR"
6170     TO WS-ERR-MSG
6171     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
6172 END-IF.
6173
6174 *
6175 IF SQLCODE = 1403
6176     MOVE 'Y' TO WS-EOC-AUTO-PREM
6177 ELSE
6178     IF     WS-CLIENT-STATE NOT EQUAL 'NY'
6179         AND WS-AUCV-AU-TYPE = 'OBEL'
6180         MOVE 'PIP' TO WS-AUCV-AU-TYPE
6181     END-IF
6182     IF WS-AUCV-AU-LIMIT-SPLIT = 'Y'
6183         IF WS-AUCV-AU-SUB-TYPE NOT EQUAL SPACES
6184             MOVE WS-AUCV-AU-SUB-TYPE TO WS-AUCV-AU-TYPE
6185         END-IF
6186     IF WS-AUCV-AU-LIMIT-SPLIT-HOLD EQUAL SPACES THEN
6187         MOVE WS-AUCV-AU-LIMIT-SPLIT TO WS-AUCV-AU-LIMIT-SPLIT-HOLD
6188         MOVE '0000000' TO WS-LIAB-LIMIT(1)
6189     END-IF
6190     MOVE "SEE END" TO WS-AUCV-AU-LIMIT
6191 END-IF
6192 PERFORM 3500-SUM-ASSM THRU 3500-EXIT
6193 END-IF.
6194
6195 *
6196 3475-EXIT.
6197 EXIT.
6198
6199 *
6200 3500-SUM-ASSM.
6201
6202 *
6203 INITIALIZE
6204     WS-ASSM-GB-AMOUNT,
6205     WS-ASSM-GB-WRIT-AMOUNT.
6206
6207 *
6208 EXEC SQL AT ORA1
6209     SELECT SUM(ASSM_GB_AMOUNT),
6210            SUM(ASSM_GB_WRIT_AMOUNT)
6211            INTO :WS-ASSM-GB-AMOUNT,
6212                :WS-ASSM-GB-WRIT-AMOUNT
6213            FROM STAS, ASSM
6214            WHERE STAS_GB_NO      = ASSM_STAS_GB_NO
6215            AND ASSM_INFC_GB_NO   = :WS-PPRT-INFC-GB-NO
6216            AND ASSM_EXIT_GB_NO   = :WS-EXIT-GB-NO
6217            AND STAS_GB_PRINT_LEVEL = 'EP'

```

```

6213      END-EXEC.
6214      *
6215      IF      SQLCODE NOT EQUAL 0
6216          AND SQLCODE NOT EQUAL 1403
6217          MOVE "*** ERROR ** FETCHING AUTO_PREM_CURS CURSOR"
6218              TO WS-ERR-MSG
6219          PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
6220      END-IF.
6221      *
6222      ADD WS-ASSM-GB-AMOUNT      TO WS-EXIT-NET-PREM.
6223      ADD WS-ASSM-GB-WRIT-AMOUNT TO WS-EXIT-WRIT-PREM.
6224      *
6225      3500-EXIT.
6226      EXIT.
6227      *
6228      * -----*
6229      * THIS PARAGRAPH PERFORMS THE FETCH FOR THE DRIVING CURSOR.      *
6230      * -----*
6231      *
6232      4000-FETCH-CURS.
6233      *
6234      *
6235      INITIALIZE
6236          WS-PPRT-POLY-GB-NO
6237          WS-PPRT-INFC-GB-NO.
6238      *
6239      EXEC SQL
6240          FETCH CURS_PRINT
6241          INTO :WS-PPRT-POLY-GB-NO,
6242              :WS-PPRT-INFC-GB-NO
6243      END-EXEC.
6244      *
6245      IF      SQLCODE NOT EQUAL 0
6246          AND SQLCODE NOT EQUAL 1403
6247          MOVE "*** ERROR ** FETCHING CURS_PRINT CURSOR" TO
6248              WS-ERR-MSG
6249          PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
6250      END-IF.
6251      *
6252      IF SQLCODE EQUAL 1403
6253          MOVE "Y" TO WS-EOC-DRIVING
6254      END-IF.
6255      *
6256      4000-EXIT.
6257      *
6258      * -----*
6259      * THIS PARAGRAPH WILL GET THE PREV INFORCE.      *
6260      * -----*
6261      *
6262      4300-GET-PREV-INFC.
6263      *
6264      INITIALIZE
6265          WS-PREV-INFC-GB-NO.
6266      *
6267      EXEC SQL AT ORA1
6268          SELECT INFC_GB_NO

```

```

6269         INTO :WS-PREV-INFC-GB-NO
6270 FROM (SELECT INFC_GB_NO
6271         FROM INFC
6272         WHERE INFC_GB_NO < :WS-PPRT-INFC-GB-NO
6273         AND INFC_HSTP_GB_CODE_OF != 'NP'
6274         AND INFC_POLY_GB_NO_PART_OF =
6275         :WS-PPRT-POLY-GB-NO
6276         ORDER BY INFC_GB_NO DESC)
6277 WHERE ROWNUM = 1
6278 END-EXEC.
6279 *
6280 IF     SQLCODE NOT EQUAL 0
6281     AND SQLCODE NOT EQUAL 1403
6282     MOVE "*** ERROR ** GETTING PREVIOUS INFORCE"
6283     TO WS-ERR-MSG
6284     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
6285 END-IF.
6286 *
6287 4300-EXIT.
6288 EXIT.
6289 *
6290 *-----*
6291 * THIS PARAGRAPH WILL RE-INITIALIZE SOME WORKING STORAGE      *
6292 * VARIABLES.                                                    *
6293 *-----*
6294 *
6295 5000-REINIT-VAR.
6296 *
6297 INITIALIZE
6298     WS-PRINT-PAGE-2A
6299     WS-CREDIT-COUNT
6300     WS-SUM-WRIT-PREM-LIAB
6301     WS-SUM-WRIT-PREM-BI
6302     WS-SUM-WRIT-PREM-PD
6303     WS-SUM-WRIT-PREM-PIP
6304     WS-SUM-WRIT-PREM-PIP
6305     WS-NY-WRIT-PREM-PIP
6306     WS-SUM-WRIT-PREM-OBEL
6307     WS-SUM-WRIT-PREM-ADDED-PIP
6308     WS-SUM-WRIT-PREM-NY-ADDED-PIP
6309     WS-SUM-WRIT-PREM-PPI
6310     WS-SUM-WRIT-PREM-MED-PAY
6311     WS-SUM-WRIT-PREM-MEDEXP
6312     WS-SUM-WRIT-PREM-UM
6313     WS-SUM-WRIT-PREM-UI
6314     WS-SUM-WRIT-PREM-COMP
6315     WS-SUM-WRIT-PREM-COLL
6316     WS-SUM-WRIT-PREM-SPEC
6317     WS-SUM-WRIT-PREM-TOW
6318     WS-SUM-WRIT-PREM-BUYBACK
6319     WS-SUM-WRIT-PREM-RENTAL
6320     WS-SUM-WRIT-PREM-HIRED
6321     WS-SUM-WRIT-PREM-NON-OWNED
6322     WS-SUM-WRIT-PREM-HIRPHY
6323     WS-ASSM-TABLE
6324     WS-KY-ASSM-TABLE

```


6325	WS-SCH-LOSS-PAYEE-TABLE
6326	WS-LOSS-PAYEE-TABLE
6327	WS-ENDORSEMENT-TABLE
6328	WS-SCHEDULE-AUTO-TABLE
6329	WS-SCHEDULE-WRIT-PREM-TABLE
6330	WS-SCHEDULE-NET-PREM-TABLE
6331	WS-VAR-PIP-EXIST
6332	WS-VAR-COMP-EXIST
6333	WS-VAR-SPEC-LOSS-EXIST
6334	WS-VAR-COLL-EXIST
6335	WS-GLASS-EXIST-L
6336	WS-GLASS-EXIST-M
6337	WS-VAR-TOW-EXIST
6338	WS-VAR-ADPIP-EXIST
6339	WS-VAR-PPI-EXIST
6340	WS-VAR-BUY-EXIST
6341	WS-VAR-MEDEXP-EXIST
6342	WS-VAR-ILB-EXIST
6343	WS-STATE-COUNT.
6344	*
6345	MOVE 1 TO WS-ENTRY-COUNT.
6346	MOVE 1 TO WS-ENTRY-COUNT-2.
6347	MOVE 1 TO WS-ENTRY-COUNT-3.
6348	MOVE 1 TO WS-ENTRY-COUNT-4.
6349	MOVE 1 TO WS-ENTRY-COUNT-5.
6350	MOVE 1 TO WS-ENTRY-COUNT-6.
6351	MOVE 1 TO WS-HIREDAUTO-COUNT.
6352	MOVE 1 TO WS-NONOWNED-COUNT.
6353	
6354	*
6355	MOVE "0000000" TO WS-LIAB-LIMIT(1).
6356	MOVE "0000000" TO WS-LIAB-LIMIT(2).
6357	MOVE "0000000" TO WS-MED-PAY-LIMIT(1).
6358	MOVE "0000000" TO WS-MED-PAY-LIMIT(2).
6359	MOVE "0000000" TO WS-UM-LIMIT(1).
6360	MOVE "0000000" TO WS-SUM-LIMIT(1).
6361	MOVE "0000000" TO WS-SUM-LIMIT(2).
6362	MOVE "0000000" TO WS-UM-LIMIT(2).
6363	MOVE "0000000" TO WS-UI-LIMIT(1).
6364	MOVE "0000000" TO WS-UI-LIMIT(2).
6365	MOVE "0000000" TO WS-PIP-DEATH-LIMIT(1).
6366	MOVE "0000000" TO WS-PIP-DEATH-LIMIT(2).
6367	MOVE "0000000" TO WS-PIP-EXP-LIMIT(1).
6368	MOVE "0000000" TO WS-PIP-EXP-LIMIT(2).
6369	MOVE "0000000" TO WS-PIP-BASIC-LIMIT(1).
6370	MOVE "0000000" TO WS-PIP-BASIC-LIMIT(2).
6371	MOVE "0000000" TO WS-NY-PIP-BASIC-LIMIT(1).
6372	MOVE "0000000" TO WS-NY-PIP-BASIC-LIMIT(2).
6373	MOVE "0000000" TO WS-PIP-WORK-LIMIT(1).
6374	MOVE "0000000" TO WS-PIP-WORK-LIMIT(2).
6375	MOVE "0000000" TO WS-OBEL-LIMIT(1).
6376	MOVE "0000000" TO WS-OBEL-LIMIT(2).
6377	MOVE "0000000" TO WS-APIP-BASIC-LIMIT(1).
6378	MOVE "0000000" TO WS-APIP-BASIC-LIMIT(2).
6379	MOVE "0000000" TO WS-NY-APIP-BASIC-LIMIT(1).
6380	MOVE "0000000" TO WS-NY-APIP-BASIC-LIMIT(2).

```

6381         MOVE "0000000" TO WS-NY-APDTH-ADD-LIMIT(1).
6382     *
6383     5000-EXIT.
6384     EXIT.
6385     *
6386     *-----*
6387     * THIS PARAGRAPH WILL PRINT THE AUTO POLICY FOR THE PPRT RECORD *
6388     * JUST PROCESSED. *
6389     *-----*
6390     *
6391     7000-PRINT-AUTO-POLICY.
6392     *
6393     PERFORM 7100-PRINT-DEC-PAGE THRU 7100-EXIT.
6394     *
6395     PERFORM 7200-PRINT-COVERAGE THRU 7200-EXIT.
6396     *
6397     PERFORM 7300-PRINT-HIRED-NON-OWNED THRU 7300-EXIT.
6398     *
6399     MOVE 'Y' TO WS-FIRST-TIME.
6400     MOVE 1 TO WS-ENTRY-COUNT.
6401     PERFORM 7400-PRINT-SCHEDULE-PAGE THRU 7400-EXIT
6402     UNTIL WS-TBL-VHDS-AU-NO(WS-ENTRY-COUNT) = 0.
6403     *
6404     MOVE 1 TO WS-ENTRY-COUNT-4.
6405     PERFORM 7500-PRINT-LOSS-PAYEE THRU 7500-EXIT
6406     UNTIL WS-SCH-LP-ACCT(WS-ENTRY-COUNT-4) = 0.
6407     *
6408     MOVE 1 TO WS-ENTRY-COUNT-3.
6409     PERFORM 7600-PRINT-LP-NOTICE THRU 7600-EXIT
6410     UNTIL WS-LP-ACCT(WS-ENTRY-COUNT-3) = 0.
6411     *
6412     MOVE SPACES TO WS-AUCV-AU-LIMIT-SPLIT-HOLD.
6413     *
6414     7000-EXIT.
6415     EXIT.
6416     *
6417     *-----*
6418     * THIS PARAGRAPH WILL PRINT THE DEC PAGE OF THE AUTO POLICY. *
6419     *-----*
6420     *
6421     7100-PRINT-DEC-PAGE.
6422     *
6423     MOVE 'AUDEC' TO WS-PLFD-GB-FORM-DESC.
6424     MOVE 'Y' TO WS-PLFD-GB-ENABLE.
6425     MOVE 'XX' TO WS-PLFD-GB-OPTION-1.
6426     MOVE 'XX' TO WS-PLFD-GB-OPTION-2.
6427     *
6428     IF WS-INFC-GB-PRIMARY-STATE NOT EQUAL SPACES THEN
6429     MOVE WS-INFC-GB-PRIMARY-STATE TO WS-PLFD-GB-STATE
6430     ELSE
6431     MOVE WS-CLIENT-STATE TO WS-PLFD-GB-STATE
6432     END-IF.
6433     *
6434     PERFORM 8500-GET-MACRO THRU 8500-EXIT.
6435     *
6436     IF SQLCODE EQUAL 1403

```

```

6437         MOVE 'XX'          TO WS-PLFD-GB-STATE
6438         PERFORM 8500-GET-MACRO THRU 8500-EXIT
6439     END-IF.
6440 *
6441     IF SQLCODE NOT EQUAL 0
6442         MOVE "ERROR SELECTING MACRO FOR STATE XX - 7100"
6443         TO WS-ERR-MSG
6444         PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
6445     END-IF.
6446 *
6447     MOVE SPACES          TO WS-PRINT-MACRO-LINE.
6448     MOVE ESC              TO WS-PRINT-ESC.
6449     MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO.
6450     MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE.
6451     WRITE PRINT-LINE AFTER ADVANCING PAGE.
6452 *
6453     MOVE SPACES      TO PRINT-LINE.
6454     MOVE WS-COURIER   TO PRT-MISC-INFO.
6455     WRITE PRINT-LINE.
6456 *
6457
6458     if WS-POLY-BSOR-GB-FM-ACCT = 5
6459     PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6460         VARYING WS-BLANK-LINE FROM 1 BY 1
6461         UNTIL WS-BLANK-LINE > 3
6462     else
6463     PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6464         VARYING WS-BLANK-LINE FROM 1 BY 1
6465         UNTIL WS-BLANK-LINE > 4
6466     end-if.
6467
6468     EXEC SQL AT ORA1
6469         SELECT COUNT(*)
6470         INTO :WS-MI-LOCTS
6471         FROM
6472             loct,
6473             coit
6474         WHERE
6475             coit_loct_gb_no      = loct_gb_no
6476             AND loct_stat_gb_alpha_state_in = 'MI'
6477             AND coit_infc_gb_no    = :WS-PPRT-INFC-GB-NO
6478     end-exec.
6479 *
6480 **HLC --CUT COUNT FROM 6 TO 4 TO ALLOW FOR NEW STUFF--
6481     IF WS-INFC-GB-PRIMARY-STATE = 'MI'
6482     OR WS-MI-LOCTS > 0 THEN
6483         MOVE SPACES TO PRINT-LINE
6484         MOVE WS-12-BOLD TO PRT-MISC-INFO
6485         WRITE PRINT-LINE
6486         MOVE WS-NOTICE-MI1 TO PRT-MISC-INFO
6487         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6488         MOVE WS-NOTICE-MI2 TO PRT-MISC-INFO
6489         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6490         MOVE SPACES      TO PRINT-LINE
6491         MOVE WS-COURIER   TO PRT-MISC-INFO
6492         WRITE PRINT-LINE AFTER ADVANCING 0 LINES

```

```

6493     ELSE
6494         MOVE SPACES TO PRINT-LINE
6495         WRITE PRINT-LINE AFTER ADVANCING 3 LINES
6496     END-IF.
6497 **HLC
6498     MOVE SPACES TO PRINT-LINE.
6499     IF WS-INFC-HSTP-GB-CODE-OF = 'RW'
6500         MOVE "          REWRITE OF : " TO PRT-MISC-INFO
6501     END-IF.
6502     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6503 *
6504
6505     if WS-POLY-BSOR-GB-FM-ACCT = 5
6506         MOVE SPACES TO PRINT-LINE
6507         MOVE WS-POLY-NO          TO PRT-POLY-NO
6508         MOVE WS-POLY-BSOR-GB-ACCOUNT TO PRT-ACCT-NO
6509         WRITE PRINT-LINE
6510     else
6511         MOVE SPACES TO PRINT-LINE
6512         MOVE WS-POLY-NO          TO PRT-POLY-NO
6513         MOVE WS-POLY-BSOR-GB-ACCOUNT TO PRT-ACCT-NO
6514         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6515     end-if.
6516
6517
6518
6519 *
6520     if WS-POLY-BSOR-GB-FM-ACCT = 5
6521         PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6522             VARYING WS-BLANK-LINE FROM 1 BY 1
6523             UNTIL WS-BLANK-LINE > 8
6524     else
6525         PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6526             VARYING WS-BLANK-LINE FROM 1 BY 1
6527             UNTIL WS-BLANK-LINE > 7
6528     end-if.
6529
6530
6531
6532
6533 *
6534     MOVE SPACES TO PRINT-LINE.
6535     MOVE WS-CLIENT-NAME TO PRT-NAME-LOCT-INFO.
6536     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6537 *
6538     MOVE SPACES TO PRINT-LINE.
6539     MOVE WS-CLIENT-ADNAME TO PRT-NAME-LOCT-INFO.
6540     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6541 *
6542     PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6543         VARYING WS-BLANK-LINE FROM 1 BY 1
6544         UNTIL WS-BLANK-LINE > 2.
6545 *
6546     MOVE SPACES TO PRINT-LINE.
6547     MOVE WS-CLIENT-STREET1 TO PRT-NAME-LOCT-INFO.
6548     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.

```

```

6549      *
6550      IF WS-CLIENT-STREET2 NOT EQUAL SPACES
6551          MOVE SPACES TO PRINT-LINE
6552          MOVE WS-CLIENT-STREET2 TO PRT-NAME-LOCT-INFO
6553          WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6554      END-IF.
6555      *
6556      MOVE SPACES TO PRINT-LINE.
6557      MOVE WS-CLIENT-CITY-ST-ZIP TO PRT-NAME-LOCT-INFO.
6558      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6559      *
6560      IF WS-CLIENT-STREET2 NOT EQUAL SPACES
6561          PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6562          VARYING WS-BLANK-LINE FROM 1 BY 1
6563          UNTIL WS-BLANK-LINE > 1
6564      ELSE
6565          PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6566          VARYING WS-BLANK-LINE FROM 1 BY 1
6567          UNTIL WS-BLANK-LINE > 2
6568      END-IF.
6569      *
6570      MOVE SPACES TO PRINT-LINE.
6571      EVALUATE WS-DAYS-POLICY-PERIOD
6572          WHEN 1
6573              STRING " ",
6574                  WS-DAYS-POLICY-PERIOD,
6575                  " DAY"
6576              DELIMITED BY SIZE
6577              INTO PRT-POLICY-INFO
6578          WHEN 365
6579              MOVE " 1 YEAR" TO PRT-POLICY-INFO
6580          WHEN 366
6581              MOVE " 1 YEAR" TO PRT-POLICY-INFO
6582          WHEN OTHER
6583              STRING " ",
6584                  WS-DAYS-POLICY-PERIOD,
6585                  " DAYS"
6586              DELIMITED BY SIZE
6587              INTO PRT-POLICY-INFO
6588      END-EVALUATE.
6589      IF WS-LOCT-GB-STREET2 NOT EQUAL SPACES
6590          WRITE PRINT-LINE AFTER ADVANCING 2 LINES
6591      ELSE
6592          WRITE PRINT-LINE AFTER ADVANCING 2 LINES
6593      END-IF.
6594      *
6595      MOVE SPACES TO PRINT-LINE.
6596      MOVE WS-POLY-GB-EFF-DATE TO PRT-POLICY-INFO.
6597      WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
6598      *
6599      MOVE SPACES TO PRINT-LINE.
6600      MOVE WS-POLY-GB-EXP-DATE TO PRT-POLICY-INFO.
6601      WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
6602      *
6603      PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6604          VARYING WS-BLANK-LINE FROM 1 BY 1

```

```

6605         UNTIL WS-BLANK-LINE > 6.
6606     *
6607     MOVE SPACES TO PRINT-LINE.
6608     STRING " ",
6609         WS-BSFM-GB-TYPE-DESC
6610         DELIMITED BY SIZE
6611     INTO PRT-POLICY-INFO.
6612     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6613     *
6614     MOVE SPACES TO PRINT-LINE.
6615     STRING " ",
6616         WS-BSTP-GB-TYPE-DESC
6617         DELIMITED BY SIZE
6618     INTO PRT-POLICY-INFO.
6619     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
6620     *
6621     *32513.6762 AAP ADDED FOR SPECIAL PA HCOLL COVERAGE WORDING
6622     IF WS-PA-EXIST = "Y"
6623     IF WS-SUM-WRIT-PREM-HIRPHY-1403 = "N" THEN
6624         MOVE SPACES TO PRINT-LINE
6625         MOVE "THIS POLICY PROVIDES COLLISION COVERAGE ON RENTAL
6626     -     " VEHICLES SUBJECT TO A " TO PRT-PA-VERB
6627         WRITE PRINT-LINE AFTER ADVANCING 3 LINES
6628     *
6629         MOVE SPACES TO PRINT-LINE
6630         STRING "LIMIT OF LIABILITY NOT TO EXCEED $",
6631             WS-HCOLL-LIMIT(1:WS-LIMIT-LEN),
6632             " FOR ONE VEHICLE FOR ONE LOSS"
6633         DELIMITED BY SIZE
6634         INTO PRT-PA-VERB
6635         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6636     *
6637         MOVE SPACES TO PRINT-LINE
6638         STRING "MINUS $",
6639             WS-HCOLL-DEDUCT(1:WS-DEDUCT-LEN),
6640             " DEDUCTIBLE FOR EACH COVERED VEHICLE."
6641         DELIMITED BY SIZE
6642         INTO PRT-PA-VERB
6643         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6644     *
6645         PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6646             VARYING WS-BLANK-LINE FROM 1 BY 1
6647             UNTIL WS-BLANK-LINE > 5
6648     ELSE
6649         MOVE SPACES TO PRINT-LINE
6650         MOVE "THIS POLICY DOES NOT PROVIDE COLLISION COVERAGE
6651     -     " ON RENTAL VEHICLES." TO PRT-PA-VERB
6652         WRITE PRINT-LINE AFTER ADVANCING 3 LINES
6653
6654         PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6655             VARYING WS-BLANK-LINE FROM 1 BY 1
6656             UNTIL WS-BLANK-LINE > 7
6657     END-IF
6658     ELSE
6659         PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6660             VARYING WS-BLANK-LINE FROM 1 BY 1

```



```

6661         UNTIL WS-BLANK-LINE > 10
6662     END-IF.
6663 *
6664     MOVE SPACES TO PRINT-LINE.
6665     MOVE WS-POLY-GB-SYSDATE TO PRT-POLICY-DATE
6666     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6667
6668
6669     MOVE SPACES TO PRINT-LINE.
6670     MOVE WS-POLY-SYSTEM-RATED-IND TO PRT-SYS-RATED-IND.
6671     WRITE PRINT-LINE AFTER ADVANCING 6 LINES.
6672
6673     MOVE SPACES TO PRINT-LINE.
6674     MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-3-INFO.
6675     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6676
6677
6678
6679
6680
6681 7100-EXIT.
6682     EXIT.
6683 *
6684 * -----*
6685 * THIS PARAGRAPH WILL PRINT PAGE 2 ITEM TWO OF THE AUTO POLICY. *
6686 * -----*
6687 *
6688 7200-PRINT-COVERAGE.
6689 *
6690     MOVE 'AUCOV'      TO WS-PLFD-GB-FORM-DESC.
6691     MOVE 'Y'          TO WS-PLFD-GB-ENABLE.
6692     MOVE 'NS'         TO WS-PLFD-GB-OPTION-1.
6693     MOVE 'S'          TO WS-PLFD-GB-OPTION-2.
6694 *
6695     IF WS-INFC-GB-PRIMARY-STATE NOT EQUAL SPACES THEN
6696         MOVE WS-INFC-GB-PRIMARY-STATE TO WS-PLFD-GB-STATE
6697     ELSE
6698         MOVE WS-CLIENT-STATE TO WS-PLFD-GB-STATE
6699     END-IF.
6700 *
6701     PERFORM 8500-GET-MACRO THRU 8500-EXIT.
6702 *
6703     IF SQLCODE = 1403
6704         MOVE 'XX'      TO WS-PLFD-GB-STATE
6705         PERFORM 8500-GET-MACRO THRU 8500-EXIT
6706     END-IF.
6707 *
6708     IF SQLCODE NOT EQUAL 0
6709         MOVE "ERROR SELECTING MACRO FOR STATE XX - 7200"
6710         TO WS-ERR-MSG
6711         PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
6712     END-IF.
6713 *
6714     MOVE SPACES      TO WS-PRINT-MACRO-LINE.
6715     MOVE ESC         TO WS-PRINT-ESC.
6716     MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO.

```

```

6717      MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE.
6718      WRITE PRINT-LINE AFTER ADVANCING PAGE.
6719  *
6720      MOVE SPACES    TO PRINT-LINE.
6721      MOVE ESC    TO FEED-ESC-8.
6722      MOVE ESC    TO COL-ESC.
6723      WRITE PRINT-LINE FROM LINEFEED-8.
6724  *
6725      IF WS-PLFD-GB-OPTION-OUT = 'NS'
6726          PERFORM 7250-PRINT-NON-SIMPLIFIED THRU 7250-EXIT
6727      ELSE
6728          MOVE SPACES    TO PRINT-LINE
6729          MOVE WS-12-CHAR TO PRT-MISC-INFO
6730          WRITE PRINT-LINE
6731          PERFORM 7225-PRINT-SIMPLIFIED    THRU 7225-EXIT
6732      END-IF.
6733  *
6734      MOVE SPACES TO PRINT-LINE.
6735      MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-4-INFO.
6736      IF WS-PRT-2A = "Y" THEN
6737          WRITE PRINT-LINE AFTER ADVANCING 4 LINES
6738      ELSE
6739          WRITE PRINT-LINE AFTER ADVANCING 3 LINES
6740      END-IF.
6741  *
6742      7200-EXIT.
6743      EXIT.
6744  *
6745  * -----*
6746  *  THIS PARAGRAPH WILL PRINT A SIMPLIFIED PAGE 2.      *
6747  * -----*
6748      7225-PRINT-SIMPLIFIED.
6749  *
6750      MOVE "N" TO WS-PRT-2A.
6751  *
6752      PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6753      VARYING WS-BLANK-LINE FROM 1 BY 1
6754      UNTIL WS-BLANK-LINE > 15.
6755  *
6756      MOVE SPACES    TO PRINT-LINE.
6757      MOVE ESC TO FEED-ESC-6.
6758      WRITE PRINT-LINE FROM LINEFEED-6.
6759  *
6760      PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
6761      VARYING WS-BLANK-LINE FROM 1 BY 1
6762      UNTIL WS-BLANK-LINE > 4.
6763  *
6764      MOVE SPACES TO PRINT-LINE.
6765      IF  WS-INFC-AU-COV-AU-LIAB NOT EQUAL SPACES
6766          OR WS-SUM-WRIT-PREM-LIAB NOT EQUAL 0
6767          MOVE "LIABILITY INSURANCE" TO PRT-COVERAGE-DESC
6768          MOVE WS-INFC-AU-COV-AU-LIAB(1:5) TO PRT-COVERAGE-AUTO
6769          IF WS-LIAB-LIMIT(1) = "SEE END"
6770              AND WS-LIAB-LIMIT(2) = "0000000"
6771      OR WS-AUCV-AU-LIMIT-SPLIT-BI = 'Y'
6772          MOVE " (SEE ENDORSEMENT)"

```

```

6773         TO PRT-COVERAGE-SEP-STATD
6774 IF WS-AUCV-AU-LIMIT-SPLIT-BI NOT EQUAL 'Y'
6775     MOVE " EACH ACCIDENT" TO PRT-COVERAGE-TEXT
6776 END-IF
6777 ELSE
6778     IF WS-LIAB-LIMIT(1) NUMERIC
6779         MOVE WS-LIAB-LIMIT(1) TO PRT-COVERAGE-LIMIT
6780     ELSE
6781         MOVE WS-LIAB-LIMIT(1) TO PRT-COVERAGE-LIMIT-CHAR
6782     END-IF
6783     IF WS-LIAB-LIMIT(2) NUMERIC
6784         IF WS-LIAB-LIMIT(2) NOT EQUAL "0000000"
6785             MOVE "$" TO PRT-DOLLAR-SIGN-2
6786         END-IF
6787         MOVE WS-LIAB-LIMIT(2) TO PRT-COVERAGE-LIMIT-2
6788     ELSE
6789         MOVE WS-LIAB-LIMIT(2)
6790         TO PRT-COVERAGE-LIMIT-2-CHAR
6791     END-IF
6792     MOVE " EACH ACCIDENT" TO PRT-COVERAGE-TEXT
6793 END-IF
6794     MOVE WS-SUM-WRIT-PREM-LIAB TO PRT-COST-DECIMAL
6795 END-IF.
6796 WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6797 ADD 1 TO WS-PG2-LINECNT.
6798 *
6799 IF WS-INFC-AU-COV-AU-LIAB(7:3) NOT EQUAL SPACES THEN
6800     IF WS-PG2-LINECNT >= 40 THEN
6801         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
6802     END-IF
6803     MOVE SPACES TO PRINT-LINE
6804     MOVE WS-INFC-AU-COV-AU-LIAB(7:3) TO PRT-COVERAGE-AUTO
6805     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6806     ADD 1 TO WS-PG2-LINECNT
6807 END-IF.
6808 *
6809 *32513.6762 AAP ADDED SSL WORDING FOR NY
6810 IF WS-PG2-LINECNT >= 38 THEN
6811     PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
6812 END-IF.
6813 IF WS-SSL-EXIST = "Y" THEN
6814     MOVE SPACES TO PRINT-LINE
6815     MOVE " SUPPLEMENTAL SPOUSAL" TO PRT-COVERAGE-DESC
6816     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6817     ADD 1 TO WS-PG2-LINECNT
6818 *
6819     MOVE SPACES TO PRINT-LINE
6820     MOVE " LIABILITY (SSL)" TO PRT-COVERAGE-DESC
6821     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6822     ADD 1 TO WS-PG2-LINECNT
6823 *
6824 END-IF.
6825 *
6826 IF WS-PG2-LINECNT >= 36 THEN
6827     PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
6828 END-IF.

```

```

6829         MOVE SPACES TO PRINT-LINE.
6830 **-----
6831         MOVE "PERSONAL INJURY PROTECTION" TO PRT-COVERAGE-DESC.
6832         MOVE WS-INFC-AU-COV-AU-PIP(1:5) TO PRT-COVERAGE-AUTO.
6833         MOVE "SEPARATELY STATED IN EACH"
6834             TO PRT-COVERAGE-SEP-STATED.
6835
6836         IF WS-VAR-PIP-EXIST = 'Y' THEN
6837             MOVE WS-SUM-WRIT-PREM-PIP TO PRT-ZERO-COST-DECIMAL
6838         ELSE
6839             MOVE WS-SUM-WRIT-PREM-PIP TO PRT-COST-DECIMAL
6840         END-IF.
6841
6842         WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
6843         ADD 2 TO WS-PG2-LINECNT.
6844 *-----
6845         MOVE SPACES TO PRINT-LINE.
6846         MOVE " (OR EQUIVALENT NO-FAULT" TO PRT-COVERAGE-DESC.
6847         MOVE WS-INFC-AU-COV-AU-PIP(7:3) TO PRT-COVERAGE-AUTO.
6848         MOVE "PIP ENDORSEMENT " TO PRT-COVERAGE-SEP-STATED.
6849
6850         WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6851         ADD 1 TO WS-PG2-LINECNT.
6852         MOVE SPACES TO PRINT-LINE.
6853         MOVE " COVERAGE)" TO PRT-COVERAGE-DESC.
6854 **-----
6855         IF WS-MI-EXIST = "Y" THEN
6856
6857             EXEC SQL AT ORA1
6858                 SELECT EXIT_GB_DEDUCTIBLE
6859                     INTO WS-PIP-DEDUCTIBLE
6860                     FROM AUTO_COVERAGE, VEHICLE_CODES,
6861                          VEHICLE_DESC, EXPOSURE, COIT
6862                     WHERE EXIT_VHCD_AU_NO = VHCD_AU_NO
6863                     AND EXIT_AUCV_AU_NO = AUCV_AU_NO
6864                     AND EXIT_GB_NO = COIT_EXIT_GB_NO
6865                     AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
6866                     AND EXIT_VHDS_AU_NO = VHDS_AU_NO
6867                     AND EXIT_VHDS_AU_NO = :WS-EXIT-VHDS-AU-NO
6868                     AND EXIT_WC_CLASS_CODE IN ('PIP')
6869             END-EXEC
6870
6871             if WS-PIP-DEDUCTIBLE > 0 then
6872                 MOVE "$" TO PRT-PIP-COV-DESC
6873                 MOVE WS-PIP-DEDUCTIBLE TO PRT-PIP-DEDUCT
6874                 MOVE " DED (See Endorsement) " TO PRT-PIP-DESC
6875             end-if
6876         END-IF.
6877
6878 **-----
6879
6880         IF WS-KY-EXIST = "Y" AND WS-PIP-DEDUCTIBLE > 0 THEN
6881             MOVE "$" TO PRT-PIP-COV-DESC
6882             MOVE WS-PIP-DEDUCTIBLE TO PRT-PIP-DEDUCT
6883
6884             IF WS-MULTISTATE = "Y" THEN

```

```

6885         MOVE " DEDUCTIBLE - KY" TO PRT-PIP-DESC
6886     ELSE
6887         MOVE " DEDUCTIBLE " TO PRT-PIP-DESC
6888     END-IF
6889
6890 END-IF.
6891
6892
6893 **-----
6894     IF WS-MI-EXIST = "Y" THEN
6895         IF WS-PIP-DEDUCTIBLE > 0 then
6896             MOVE "$" TO PRT-PIP-COV-DESC
6897             MOVE WS-PIP-DEDUCTIBLE TO PRT-PIP-DEDUCT
6898             MOVE " DED (See Endorsement) " TO PRT-PIP-DESC
6899         end-if
6900     END-IF.
6901 **-----
6902     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6903     ADD 1 TO WS-PG2-LINECNT.
6904 *
6905     PERFORM 7230-PRT-ADD-PIP   THRU 7230-EXIT
6906 *
6907     IF WS-VA-EXIST = "Y" THEN
6908         IF WS-PG2-LINECNT >= 35 THEN
6909             PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
6910         END-IF
6911         MOVE SPACES TO PRINT-LINE
6912         MOVE "MEDICAL EXPENSE AND" TO PRT-COVERAGE-DESC
6913         MOVE "SEPARATELY STATED IN EACH"
6914             TO PRT-COVERAGE-SEP-STATED
6915         MOVE WS-INFC-AU-COV-AU-MED-EXP(1:5)
6916             TO PRT-COVERAGE-AUTO
6917 *
6918         IF WS-VAR-MEDEXP-EXIST = 'Y' OR
6919             WS-VAR-ILB-EXIST = 'Y' THEN
6920             MOVE WS-SUM-WRIT-PREM-MEDEXP
6921                 TO PRT-ZERO-COST-DECIMAL
6922         ELSE
6923             MOVE WS-SUM-WRIT-PREM-MEDEXP TO PRT-COST-DECIMAL
6924         END-IF
6925         WRITE PRINT-LINE AFTER ADVANCING 2 LINES
6926         ADD 2 TO WS-PG2-LINECNT
6927 *
6928         MOVE SPACES TO PRINT-LINE
6929         MOVE " INCOME LOSS BENEFITS" TO PRT-COVERAGE-DESC
6930         MOVE "MEDICAL EXPENSE AND INCOME"
6931             TO PRT-COVERAGE-SEP-STATED
6932         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6933         ADD 1 TO WS-PG2-LINECNT
6934 *
6935         MOVE SPACES TO PRINT-LINE
6936         IF WS-MULTISTATE = "Y" THEN
6937             MOVE " (VA ONLY)" TO PRT-COVERAGE-DESC
6938         END-IF
6939         MOVE "LOSS BENEFITS ENDORSEMENT"
6940             TO PRT-COVERAGE-SEP-STATED

```

```

6941         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6942         ADD 1 TO WS-PG2-LINECNT
6943     END-IF.
6944 *
6945     MOVE SPACES TO PRINT-LINE.
6946     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
6947     ADD 1 TO WS-PG2-LINECNT.
6948 *
6949     IF    WS-INFC-AU-COV-AU-MED-PAY  NOT EQUAL SPACES
6950     OR    WS-SUM-WRIT-PREM-MED-PAY  NOT EQUAL 0
6951         IF WS-PG2-LINECNT >= 38 THEN
6952             PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
6953         END-IF
6954         MOVE SPACES TO PRINT-LINE
6955         MOVE "AUTO MEDICAL PAYMENTS" TO PRT-COVERAGE-DESC
6956         MOVE WS-INFC-AU-COV-AU-MED-PAY(1:5)
6957         TO PRT-COVERAGE-AUTO
6958         display 'MED AMOUNT: ' PRT-COVERAGE-AUTO
6959         IF    WS-MED-PAY-LIMIT(1) = "SEE END"
6960             AND WS-MED-PAY-LIMIT(2) = "0000000"
6961             MOVE " (SEE ENDORSEMENT)"
6962             TO PRT-COVERAGE-SEP-STATED
6963         ELSE
6964             IF WS-MED-PAY-LIMIT(1) NUMERIC
6965                 MOVE WS-MED-PAY-LIMIT(1) TO PRT-COVERAGE-LIMIT
6966             ELSE
6967                 MOVE WS-MED-PAY-LIMIT(1)
6968                 TO PRT-COVERAGE-LIMIT-CHAR
6969             END-IF
6970             IF WS-MED-PAY-LIMIT(2) NUMERIC
6971                 IF WS-MED-PAY-LIMIT(2) NOT EQUAL "0000000"
6972                     MOVE "$"      TO PRT-DOLLAR-SIGN-2
6973                 ELSE
6974                     MOVE SPACES    TO PRT-DOLLAR-SIGN-2
6975                 END-IF
6976                 MOVE WS-MED-PAY-LIMIT(2)
6977                 TO PRT-COVERAGE-LIMIT-2
6978             ELSE
6979                 MOVE WS-MED-PAY-LIMIT(2)
6980                 TO PRT-COVERAGE-LIMIT-2-CHAR
6981             END-IF
6982             MOVE " EACH INSURED" TO PRT-COVERAGE-TEXT
6983         END-IF
6984
6985 *****-- PLI32502.16393
6986         IF WS-SUM-WRIT-PREM-MED-PAY NOT EQUAL 0
6987         MOVE WS-SUM-WRIT-PREM-MED-PAY TO PRT-COST-DECIMAL
6988         ELSE
6989         MOVE WS-SUM-WRIT-PREM-MED-PAY TO
6990             PRT-ZERO-COST-DECIMAL
6991         END-IF
6992 *****-- PLI32502.16393
6993
6994
6995         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
6996         ADD 1 TO WS-PG2-LINECNT

```



```

6997     END-IF.
6998 *
6999     IF WS-INFC-AU-COV-AU-MED-PAY(7:3) NOT EQUAL SPACES THEN
7000         MOVE SPACES TO PRINT-LINE
7001         MOVE WS-INFC-AU-COV-AU-MED-PAY(7:3)
7002             TO PRT-COVERAGE-AUTO
7003         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7004         ADD 1 TO WS-PG2-LINECNT
7005     END-IF.
7006 *
7007     IF WS-NY-EXIST = "N" OR WS-STUM-EXIST = "Y"
7008         OR (WS-NY-EXIST = "Y" AND WS-MULTISTATE = "Y") THEN
7009         IF WS-PG2-LINECNT >= 37 THEN
7010             PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7011         END-IF
7012         MOVE SPACES TO PRINT-LINE
7013         MOVE "UNINSURED MOTORISTS" TO PRT-COVERAGE-DESC
7014         IF WS-INFC-AU-COV-AU-UM NOT EQUAL SPACES
7015             OR WS-SUM-WRIT-PREM-UM NOT EQUAL 0
7016             MOVE WS-INFC-AU-COV-AU-UM(1:5) TO PRT-COVERAGE-AUTO
7017         IF WS-AUCV-AU-LIMIT-SPLIT-UM = 'Y'
7018 * WS-AUCV-AU-LIMIT-SPLIT-UM = ""
7019             MOVE WS-UM-LIMIT-HOLD TO WS-AUCV-AU-SPLIT-LIMIT-2
7020             PERFORM 7685-FORMAT-SPLIT-LIMIT-2 THROUGH 7685-EXIT
7021             MOVE WS-AUCV-AU-SPLIT-LIMIT-2
7022                 TO PRT-COVERAGE-SEP-STATED
7023         ELSE
7024             IF WS-UM-LIMIT(1) = "SEE END"
7025                 AND WS-UM-LIMIT(2) = "0000000"
7026                 MOVE " (SEE ENDORSEMENT)"
7027                     TO PRT-COVERAGE-SEP-STATED
7028         ELSE
7029 *WS-UM-LIMIT(1) = 0000000
7030             IF WS-UM-LIMIT(1) NUMERIC
7031                 MOVE WS-UM-LIMIT(1) TO PRT-COVERAGE-LIMIT
7032             ELSE
7033                 MOVE WS-UM-LIMIT(1)
7034                     TO PRT-COVERAGE-LIMIT-CHAR
7035             END-IF
7036 *WS-UM-LIMIT(2) = 0000000
7037             IF WS-UM-LIMIT(2) NUMERIC
7038                 IF WS-UM-LIMIT(2) NOT EQUAL "0000000"
7039                     MOVE "$" TO PRT-DOLLAR-SIGN-2
7040                 MOVE WS-UM-LIMIT(2) TO PRT-COVERAGE-LIMIT-2
7041             END-IF
7042         ELSE
7043             MOVE WS-UM-LIMIT(2)
7044                 TO PRT-COVERAGE-LIMIT-2-CHAR
7045             END-IF
7046             MOVE " EACH ACCIDENT" TO PRT-COVERAGE-TEXT
7047         END-IF
7048     END-IF
7049     MOVE WS-SUM-WRIT-PREM-UM TO PRT-COST-DECIMAL
7050     END-IF
7051 *
7052     FOR WA UIPD SAVE/BLANK LIMIT/PX VALUES PRIOR WRITE
7053     IF WS-WA-UIPD-EXIST = "Y" THEN

```

```

7053         MOVE PRT-COVERAGE-LIMIT TO WS-WA-HOLD-LIMIT
7054         MOVE PRT-COST-DECIMAL  TO WS-WA-HOLD-PX
7055         MOVE SPACES            TO PRT-COVERAGE-LIMIT-A
7056         MOVE SPACES            TO PRT-COVERAGE-COST
7057     END-IF
7058     WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7059     ADD 2 TO WS-PG2-LINECNT
7060 END-IF.
7061 *
7062 IF WS-NC-EXIST = "Y" THEN
7063     EXEC SQL AT ORA1
7064         SELECT EXIT_GB_WRIT_PREM,
7065             TO_NUMBER(AUCV_AU_LIMIT)
7066             INTO :WS-NCRB-WRIT-PREM,
7067             :WS-NCRB-LIMIT
7068         FROM EXIT, AUCV
7069         WHERE EXIT_GB_NO IN
7070             (SELECT COIT_EXIT_GB_NO
7071              FROM COIT
7072              WHERE COIT_INFC_GB_NO
7073                = :WS-PPRT-INFC-GB_NO)
7074         AND EXIT_WC_CLASS_CODE IN ('PUM','PUMUI')
7075         AND EXIT_AUCV_AU_NO = AUCV_AU_NO
7076     END-EXEC
7077 *
7078 IF SQLCODE NOT EQUAL 0
7079     AND SQLCODE NOT EQUAL 1403
7080     MOVE "*** ERROR ** SELECTING NCRB PREM/LIMIT"
7081     TO WS-ERR-MSG
7082     PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
7083 END-IF
7084 *
7085 IF WS-NCRB-WRIT-PREM NOT EQUAL 0 THEN
7086     IF WS-PG2-LINECNT >= 40 THEN
7087         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7088     END-IF
7089     MOVE SPACES TO PRINT-LINE
7090     MOVE "NCRB UNINSURED MOTORISTS" TO PRT-COVERAGE-DESC
7091     MOVE WS-INFC-AU-COV-AU-UM(1:5) TO PRT-COVERAGE-AUTO
7092     MOVE WS-NCRB-LIMIT TO PRT-COVERAGE-LIMIT
7093     MOVE " EACH ACCIDENT" TO PRT-COVERAGE-TEXT
7094 *
7095     MOVE WS-NCRB-WRIT-PREM TO PRT-COST-DECIMAL
7096     COMPUTE WS-SUM-WRIT-PREM-UM = WS-SUM-WRIT-PREM-UM +
7097         WS-NCRB-WRIT-PREM
7098     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7099     ADD 1 TO WS-PG2-LINECNT
7100 END-IF
7101 END-IF.
7102 *
7103 IF WS-INFC-AU-COV-AU-UM(7:3) NOT EQUAL SPACES THEN
7104     MOVE SPACES TO PRINT-LINE
7105     MOVE WS-INFC-AU-COV-AU-UM(7:3) TO PRT-COVERAGE-AUTO
7106     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7107     ADD 1 TO WS-PG2-LINECNT
7108 END-IF.

```

```

7109 *
7110 IF WS-PG2-LINECNT >= 35 THEN
7111     PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7112 END-IF.
7113 IF WS-NY-EXIST = "N" OR
7114     (WS-NY-EXIST = "Y" AND WS-MULTISTATE = "Y") THEN
7115     MOVE SPACES TO PRINT-LINE
7116     MOVE "UNDERINSURED MOTORISTS" TO PRT-COVERAGE-DESC
7117     IF WS-INFC-AU-COV-AU-UI NOT EQUAL SPACES
7118         OR WS-SUM-WRIT-PREM-UI NOT EQUAL 0
7119         MOVE WS-INFC-AU-COV-AU-UI(1:5)
7120         TO PRT-COVERAGE-AUTO
7121     IF WS-AUCV-AU-LIMIT-SPLIT-UI = 'Y'
7122         MOVE WS-UI-LIMIT-HOLD TO WS-AUCV-AU-SPLIT-LIMIT-2
7123         PERFORM 7685-FORMAT-SPLIT-LIMIT-2 THROUGH 7685-EXIT
7124         MOVE WS-AUCV-AU-SPLIT-LIMIT-2
7125         TO PRT-COVERAGE-SEP-STATED
7126     ELSE
7127         IF WS-UI-LIMIT(1) = "SEE END"
7128             AND WS-UI-LIMIT(2) = "0000000"
7129             MOVE " (SEE ENDORSEMENT)"
7130             TO PRT-COVERAGE-SEP-STATED
7131         ELSE
7132             IF WS-UI-LIMIT(1) NUMERIC
7133                 MOVE WS-UI-LIMIT(1) TO PRT-COVERAGE-LIMIT
7134             ELSE
7135                 MOVE WS-UI-LIMIT(1)
7136                 TO PRT-COVERAGE-LIMIT-CHAR
7137             END-IF
7138             IF WS-UI-LIMIT(2) NUMERIC
7139                 IF WS-UI-LIMIT(2) NOT EQUAL "0000000"
7140                     MOVE "$" TO PRT-DOLLAR-SIGN-2
7141                 END-IF
7142                 MOVE WS-UI-LIMIT(2) TO PRT-COVERAGE-LIMIT-2
7143             ELSE
7144                 MOVE WS-UI-LIMIT(2)
7145                 TO PRT-COVERAGE-LIMIT-2-CHAR
7146             END-IF
7147             MOVE " EACH ACCIDENT" TO PRT-COVERAGE-TEXT
7148         END-IF
7149     END-IF
7150     MOVE WS-SUM-WRIT-PREM-UI TO PRT-COST-DECIMAL
7151 END-IF
7152 * -HERE CK UIPD
7153 IF WS-WA-UIPD-EXIST = "Y" THEN
7154     PERFORM 7226-PATCH-WA THRU 7226-EXIT
7155 ELSE
7156     WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7157     ADD 2 TO WS-PG2-LINECNT
7158 *
7159     MOVE SPACES TO PRINT-LINE
7160     MOVE " (WHEN NOT INCLUDED IN" TO PRT-COVERAGE-DESC
7161     MOVE WS-INFC-AU-COV-AU-UI(7:3) TO PRT-COVERAGE-AUTO
7162     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7163     ADD 1 TO WS-PG2-LINECNT
7164     MOVE SPACES TO PRINT-LINE

```

```

7165      MOVE " UNINSURED MOTORISTS "      TO PRT-COVERAGE-DESC
7166      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7167      ADD 1 TO WS-PG2-LINECNT
7168      MOVE SPACES              TO PRINT-LINE
7169      MOVE " COVERAGE)"        TO PRT-COVERAGE-DESC
7170      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7171      ADD 1 TO WS-PG2-LINECNT
7172      END-IF
7173      *
7174      END-IF.
7175      *
7176      IF WS-PG2-LINECNT >= 33 THEN
7177          PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7178      END-IF.
7179      IF WS-NY-EXIST = "Y" AND WS-NY-SUM-EXIST = "Y" THEN
7180          MOVE SPACES TO PRINT-LINE
7181          MOVE "SUPPLEMENTARY UNINSURED/" TO PRT-COVERAGE-DESC
7182          IF  WS-INFC-AU-COV-AU-UM NOT EQUAL SPACES
7183              OR WS-SUM-WRIT-PREM-SUM NOT EQUAL 0
7184              MOVE WS-INFC-AU-COV-AU-UM(1:5)
7185                  TO PRT-COVERAGE-AUTO
7186          IF WS-AUCV-AU-LIMIT-SPLIT-SUM = 'Y'
7187              MOVE WS-SUM-LIMIT-HOLD TO WS-AUCV-AU-SPLIT-LIMIT-2
7188              PERFORM 7685-FORMAT-SPLIT-LIMIT-2 THROUGH 7685-EXIT
7189              MOVE WS-AUCV-AU-SPLIT-LIMIT-2
7190                  TO PRT-COVERAGE-SEP-STATED
7191          ELSE
7192              IF WS-SUM-LIMIT(1) = "SEE END"
7193                  AND WS-SUM-LIMIT(2) = "0000000"
7194                  MOVE " (SEE ENDORSEMENT)"
7195                      TO PRT-COVERAGE-SEP-STATED
7196              ELSE
7197                  IF WS-SUM-LIMIT(1) NUMERIC
7198                      MOVE WS-SUM-LIMIT(1) TO PRT-COVERAGE-LIMIT
7199                  ELSE
7200                      MOVE WS-SUM-LIMIT(1)
7201                          TO PRT-COVERAGE-LIMIT-CHAR
7202                  END-IF
7203                  IF WS-SUM-LIMIT(2) NUMERIC
7204                      IF WS-SUM-LIMIT(2) NOT EQUAL "0000000"
7205                          MOVE "$"      TO PRT-DOLLAR-SIGN-2
7206                          MOVE WS-SUM-LIMIT(2)
7207                              TO PRT-COVERAGE-LIMIT-2
7208                      END-IF
7209                  ELSE
7210                      MOVE WS-SUM-LIMIT(2)
7211                          TO PRT-COVERAGE-LIMIT-2-CHAR
7212                      END-IF
7213                  MOVE " EACH ACCIDENT" TO PRT-COVERAGE-TEXT
7214              END-IF
7215          END-IF
7216          MOVE WS-SUM-WRIT-PREM-SUM  TO PRT-COST-DECIMAL
7217          MOVE 0 TO WS-SUM-WRIT-PREM-SUM
7218      END-IF
7219      WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7220      ADD 2 TO WS-PG2-LINECNT

```

```

7221 *
7222 MOVE SPACES TO PRINT-LINE
7223 MOVE " UNDERINSURED MOTORISTS (SUM)" TO PRT-COVERAGE-DESC
7224 MOVE WS-INFC-AU-COV-AU-UM(7:3) TO PRT-COVERAGE-AUTO
7225 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7226 ADD 1 TO WS-PG2-LINECNT
7227 IF WS-MULTISTATE = "Y" THEN
7228 MOVE SPACES TO PRINT-LINE
7229 MOVE " (NY ONLY)" TO PRT-COVERAGE-DESC
7230 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7231 ADD 1 TO WS-PG2-LINECNT
7232 END-IF
7233 MOVE SPACES TO PRINT-LINE
7234 MOVE WS-LINEPTR TO PRT-MISC-INFO
7235 WRITE PRINT-LINE AFTER ADVANCING 0 LINES
7236 *
7237 MOVE SPACES TO PRINT-LINE
7238 MOVE "*THE MAXIMUM AMOUNT PAYABLE UNDER THE SUM COVERAGE "
7239 TO PRT-SUM-DESCRIPTION1
7240 MOVE
7241 "SHALL BE THE POLICY'S SUM LIMIT, REDUCED AND THUS OFFSET"
7242 TO PRT-SUM-DESCRIPTION2
7243 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7244 ADD 1 TO WS-PG2-LINECNT
7245 *
7246 MOVE SPACES TO PRINT-LINE
7247 MOVE "BY MOTOR VEHICLE BODILY INJURY LIABILITY INSURANCE
7248 - " POLICY OF BOND PAYMENTS RECEIVED FROM, OR ON BEHALF OF,"
7249 TO PRT-SUM-DESCRIPTION
7250 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7251 ADD 1 TO WS-PG2-LINECNT
7252 *
7253 MOVE SPACES TO PRINT-LINE
7254 MOVE "ANY NEGLIGENT PARTY INVOLVED IN THE ACCIDENT, AS
7255 - " SPECIFIED IN THE SUM ENDORSEMENT."
7256 TO PRT-SUM-DESCRIPTION
7257 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7258 ADD 1 TO WS-PG2-LINECNT
7259 *
7260 MOVE SPACES TO PRINT-LINE
7261 MOVE WS-12-CHAR TO PRT-MISC-INFO
7262 WRITE PRINT-LINE AFTER ADVANCING 0 LINES
7263 END-IF.
7264 *
7265 IF WS-PG2-LINECNT >= 28 THEN
7266 PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7267 END-IF.
7268 MOVE SPACES TO PRINT-LINE.
7269 MOVE "PHYSICAL DAMAGE INS.:" TO PRT-COVERAGE-DESC.
7270 WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
7271 ADD 2 TO WS-PG2-LINECNT.
7272 MOVE SPACES TO PRINT-LINE.
7273 MOVE " COMPREHENSIVE COVERAGE" TO PRT-COVERAGE-DESC.
7274 MOVE WS-INFC-AU-COV-AU-COMP(1:5) TO PRT-COVERAGE-AUTO.
7275 *****
7276 MOVE "ACV OR COST OF REPAIR " TO PRT-COVERAGE-SEP-STATED.

```

```

7277 IF WS-VAR-COMP-EXIST = 'Y' THEN
7278     MOVE WS-SUM-WRIT-PREM-COMP TO PRT-ZERO-COST-DECIMAL
7279 ELSE
7280     MOVE WS-SUM-WRIT-PREM-COMP TO PRT-COST-DECIMAL
7281 END-IF.
7282 WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
7283 ADD 1 TO WS-PG2-LINECNT.
7284 *
7285     MOVE SPACES TO PRINT-LINE.
7286     MOVE WS-INFC-AU-COV-AU-COMP(7:3) TO PRT-COVERAGE-AUTO.
7287     MOVE "MINUS ANY DEDUCTIBLE"
7288         TO PRT-COVERAGE-SEP-STATED.
7289     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
7290     ADD 1 TO WS-PG2-LINECNT.
7291 *
7292     MOVE SPACES TO PRINT-LINE.
7293     MOVE " SPECIFIED CAUSES OF" TO PRT-COVERAGE-DESC.
7294     MOVE WS-INFC-AU-COV-AU-SPECIFIED(1:5) TO PRT-COVERAGE-AUTO.
7295 *****
7296     MOVE "ACV OR COST OF REPAIR " TO PRT-COVERAGE-SEP-STATED.
7297     IF WS-VAR-SPEC-LOSS-EXIST = 'Y' THEN
7298         MOVE WS-SUM-WRIT-PREM-SPEC TO PRT-ZERO-COST-DECIMAL
7299     ELSE
7300         MOVE WS-SUM-WRIT-PREM-SPEC TO PRT-COST-DECIMAL
7301     END-IF.
7302     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
7303     ADD 2 TO WS-PG2-LINECNT.
7304 *
7305     MOVE SPACES TO PRINT-LINE.
7306     MOVE " LOSS COVERAGE" TO PRT-COVERAGE-DESC.
7307     MOVE WS-INFC-AU-COV-AU-SPECIFIED(7:3) TO PRT-COVERAGE-AUTO.
7308     MOVE "MINUS ANY DEDUCTIBLE "
7309         TO PRT-COVERAGE-SEP-STATED.
7310     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
7311     ADD 1 TO WS-PG2-LINECNT.
7312 *
7313     MOVE SPACES TO PRINT-LINE.
7314     MOVE " COLLISION COVERAGE" TO PRT-COVERAGE-DESC.
7315     MOVE WS-INFC-AU-COV-AU-COLLISION(1:5) TO PRT-COVERAGE-AUTO.
7316 *****
7317     MOVE "ACV OR COST OF REPAIR " TO PRT-COVERAGE-SEP-STATED.
7318     IF WS-VAR-COLL-EXIST = 'Y' THEN
7319         MOVE WS-SUM-WRIT-PREM-COLL TO PRT-ZERO-COST-DECIMAL
7320     ELSE
7321         MOVE WS-SUM-WRIT-PREM-COLL TO PRT-COST-DECIMAL
7322     END-IF.
7323     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
7324     ADD 2 TO WS-PG2-LINECNT.
7325 *
7326     MOVE SPACES TO PRINT-LINE.
7327     MOVE WS-INFC-AU-COV-AU-COLLISION(7:3) TO PRT-COVERAGE-AUTO.
7328     MOVE "MINUS ANY DEDUCTIBLE "
7329         TO PRT-COVERAGE-SEP-STATED.
7330     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
7331     ADD 1 TO WS-PG2-LINECNT.
7332 *

```



```

7333     MOVE SPACES TO PRINT-LINE.
7334     MOVE " TOWING AND LABOR"          TO PRT-COVERAGE-DESC.
7335     MOVE WS-INFC-AU-COV-AU-TOW-LABOR(1:5) TO PRT-COVERAGE-AUTO.
7336 *****
7337     MOVE "$50 FOR EACH DISABLEMENT OF A "
7338         TO PRT-COVERAGE-SEP-STATED.
7339     IF WS-VAR-TOW-EXIST = 'Y' THEN
7340         MOVE WS-SUM-WRIT-PREM-TOW TO PRT-ZERO-COST-DECIMAL
7341     ELSE
7342         MOVE WS-SUM-WRIT-PREM-TOW TO PRT-COST-DECIMAL
7343     END-IF.
7344     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
7345     ADD 2 TO WS-PG2-LINECNT.
7346 *
7347     MOVE SPACES          TO PRINT-LINE.
7348     MOVE WS-INFC-AU-COV-AU-TOW-LABOR(7:3)
7349     TO PRT-COVERAGE-AUTO.
7350     MOVE "PRIVATE PASSENGER TYPE AUTO"
7351     TO PRT-COVERAGE-SEP-STATED.
7352     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
7353     ADD 1 TO WS-PG2-LINECNT.
7354
7355
7356 *-----
7357 *TXD Death Supp.
7358
7359     PERFORM 7305-DETERMINE-TXSD
7360     UNTIL END-OF-TXSD.
7361
7362 *-----
7363 *
7364     IF WS-INFC-AU-COV-AU-RENTAL NOT EQUAL SPACES
7365     IF WS-PG2-LINECNT >= 38 THEN
7366         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7367     END-IF
7368     MOVE SPACES TO PRINT-LINE
7369     MOVE "RENTAL REIMBURSEMENT " TO PRT-COVERAGE-DESC
7370     MOVE WS-INFC-AU-COV-AU-RENTAL(1:5) TO PRT-COVERAGE-AUTO
7371     MOVE WS-SUM-WRIT-PREM-RENTAL TO PRT-ZERO-COST-DECIMAL
7372     WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7373     ADD 2 TO WS-PG2-LINECNT
7374     MOVE SPACES TO PRINT-LINE
7375     MOVE " COVERAGE " TO PRT-COVERAGE-DESC
7376     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7377     ADD 1 TO WS-PG2-LINECNT
7378     END-IF.
7379 *
7380     IF WS-INFC-AU-COV-AU-UI(6:4) NOT EQUAL SPACES
7381     IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
7382         DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
7383         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
7384     ELSE
7385         MOVE "*ERROR* UNABLE TO PRINT ALL AUTO SYMBOLS FOR UI."
7386         TO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
7387         ADD 1 TO WS-ENTRY-COUNT-6
7388     END-IF

```

```

7389      END-IF.
7390      *
7391      PERFORM 7235-PRT-STATE      THRU 7235-EXIT.
7392      *
7393      MOVE SPACES TO PRINT-LINE.
7394      *
7395      EXEC SQL AT ORA1
7396          OPEN OTHER_COV_CURS
7397      END-EXEC.
7398
7399      IF      SQLCODE NOT EQUAL 0
7400          AND SQLCODE NOT EQUAL 1403
7401          MOVE "*** ERROR ** OPENING OTHER_COV_CURS CUROS"
7402              TO WS-ERR-MSG
7403          PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
7404      END-IF.
7405
7406      PERFORM 2395-FETCH-OTHER-COV THRU 2395-EXIT.
7407      *
7408      PERFORM 2396-PROCESS-OTHER-COV THRU 2396-EXIT
7409          UNTIL WS-EOC-OTHER-COV = 'Y'.
7410      *
7411      INITIALIZE
7412          WS-ASSM-DESC-2,
7413          WS-ASSM-WRIT.
7414      *
7415      EXEC SQL AT ORA1
7416          SELECT SUBSTR(MCTB_GB_DESC,1,60),
7417              SUM(NVL(EXIT_GB_NET_PREM,0))
7418          INTO :WS-ASSM-DESC-TXT,
7419              :WS-ASSM-WRIT
7420          FROM MCTB, EXIT
7421          WHERE MCTB_GB_CODE_TYPE = 'AUCOV'
7422              AND (MCTB_GB_EDIT IN ('FCTPP')
7423              AND EXIT_WC_CLASS_CODE IN ('TUI','TSLL','TMED',
7424                  'TUM','TPIP','TSTUM')
7425              OR (MCTB_GB_EDIT IN ('FCRPP')
7426              AND EXIT_WC_CLASS_CODE IN ('RSLL','RMED','RPIP','RUM',
7427                  'RSTUM','RFSLL','RFMED',
7428                  'RFPIP','RFUM','RFSTUM',
7429                  'RMSLL','RMMED','RMPPIP',
7430                  'RMUM','RMSTUM','RPSLL',
7431                  'RPPPIP','RPUM'))))
7432          AND EXIT_GB_NO IN
7433              (SELECT COIT_EXIT_GB_NO
7434              FROM COIT
7435              WHERE COIT_INFC_GB_NO
7436                  = :WS-PPRT-INFC-GB-NO)
7437          GROUP BY SUBSTR(MCTB_GB_DESC,1,60)
7438      END-EXEC.
7439
7440      *
7441      IF      SQLCODE NOT EQUAL 0
7442          AND SQLCODE NOT EQUAL 1403
7443          MOVE "*** ERROR ** SELECTING CT COVERAGE"
7444              TO WS-ERR-MSG

```

```

7445         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
7446     END-IF.
7447 *
7448     EXEC SQL AT ORA1
7449         SELECT DISTINCT LOCT_STAT_GB_ALPHA_STATE_IN
7450             INTO WS-ASSM-STATE
7451             FROM LOCT, COIT, EXIT
7452             WHERE EXIT_WC_CLASS_CODE
7453                 IN('TUI','TSLL','TMED','TUM','TPIP','TSTUM',
7454                   'RSLL','RMED','RPIP','RUM','RSTUM',
7455                   'RFSLL','RFMED','RFPIP','RFUM','RFSTUM',
7456                   'RMSLL','RMMED','RMPIP','RMUM','RMSTUM',
7457                   'RPSLL','RPPIP','RPUM')
7458             AND EXIT_GB_NO = COIT_EXIT_GB_NO
7459             AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
7460             AND COIT_LOCT_GB_NO = LOCT_GB_NO
7461     END-EXEC.
7462 *
7463     IF     SQLCODE NOT EQUAL 0
7464         AND SQLCODE NOT EQUAL 1403
7465         MOVE "*** ERROR ** SELECTING STATE"
7466         TO WS-ERR-MSG
7467         PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
7468     END-IF.
7469 *
7470     IF SQLCODE = 0 THEN
7471         IF WS-PG2-LINECNT > 39 THEN
7472             PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7473         END-IF
7474         MOVE WS-ASSM-DESC TO
7475             PRT-COVERAGE-DESC
7476         IF WS-ASSM-WRIT > 0 THEN
7477             MOVE WS-ASSM-WRIT
7478             TO PRT-COST-DECIMAL
7479         ELSE
7480             MOVE WS-ASSM-WRIT
7481             TO PRT-ZERO-COST-DECIMAL
7482         END-IF
7483         WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7484         ADD 2 TO WS-PG2-LINECNT
7485     END-IF.
7486 *
7487     MOVE SPACES TO WS-MCTB-EDIT.
7488     MOVE SPACES TO WS-ASSM-HOLD.
7489 *
7490     MOVE 'C' TO WS-MCTB-EDIT.
7491     MOVE ZERO TO WS-COUNT-ASSM
7492     PERFORM 2390-GET-POLICY-ASSM-INFO THRU 2390-EXIT.
7493     IF WS-COUNT-ASSM > 0 THEN
7494         COMPUTE WS-ASSM-CNT = 39 - WS-COUNT-ASSM
7495         IF WS-PG2-LINECNT >= WS-ASSM-CNT THEN
7496             PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7497         END-IF
7498     END-IF.
7499     MOVE 1 TO WS-WHAT-SUB.
7500     MOVE 0 TO WS-ASSM-COUNTER.

```

```

7501      PERFORM UNTIL WS-ASSM-COUNTER = WS-COUNT-ASSM
7502      MOVE SPACES TO PRINT-LINE
7503      IF WS-ASSM-GB-ASSMNT-CODE-2A(WS-WHAT-SUB) = "LERO"
7504      OR WS-ASSM-GB-ASSMNT-CODE-2A(WS-WHAT-SUB) = "FENAM"
7505      OR WS-ASSM-GB-ASSMNT-CODE-2A(WS-WHAT-SUB) = "FEDES"
7506      THEN
7507          MOVE WS-ASDC-GB-DESC-2A(WS-WHAT-SUB)
7508          TO PRT-COVERAGE-DESC
7509          display PRT-COVERAGE-DESC
7510
7511      ELSE
7512          MOVE WS-ASDC-GB-DESC-2A(WS-WHAT-SUB)
7513          TO PRT-COVERAGE-DESC
7514          display PRT-COVERAGE-DESC
7515      END-IF
7516      IF WS-ASDC-GB-DESC-2A(WS-WHAT-SUB) EQUAL
7517      'TRAILER INTERCHANGE'
7518      MOVE " (SEE ENDORSEMENT)"
7519      TO PRT-COVERAGE-SEP-STATD
7520      END-IF
7521      IF WS-ASSM-GB-WRIT-AMOUNT-2A(WS-WHAT-SUB) > 0 THEN
7522          MOVE WS-ASSM-GB-WRIT-AMOUNT-2A(WS-WHAT-SUB)
7523          TO PRT-COST-DECIMAL
7524      ELSE
7525          MOVE WS-ASSM-GB-WRIT-AMOUNT-2A(WS-WHAT-SUB)
7526          TO PRT-ZERO-COST-DECIMAL
7527      END-IF
7528      WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7529      ADD 1 TO WS-ASSM-COUNTER
7530      ADD 1 TO WS-WHAT-SUB
7531      ADD 2 TO WS-PG2-LINECNT
7532      END-PERFORM.
7533  *
7534      MOVE ZERO TO WS-WHAT-SUB.
7535      MOVE ZERO TO WS-ASSM-COUNTER.
7536      MOVE ZERO TO WS-COUNT-ASSM.
7537      MOVE 'S' TO WS-MCTB-EDIT.
7538      PERFORM 2390-GET-POLICY-ASSM-INFO THRU 2390-EXIT.
7539  *
7540      IF WS-ASSM-GB-AMOUNT-POLY > 0 OR
7541      (WS-NY-EXIST = "Y" AND WS-SUM-ASSM-MVFEE-1403 = 'N') THEN
7542          COMPUTE WS-ASSM-CNT = 36 - (WS-COUNT-ASSM + 1)
7543          IF WS-PG2-LINECNT >= WS-ASSM-CNT THEN
7544              PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7545          END-IF
7546          MOVE SPACES TO PRINT-LINE
7547          MOVE "SURCHARGES, TAXES AND FEES "
7548          TO PRT-2A-COVERAGE-DESC
7549          WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7550          ADD 2 TO WS-PG2-LINECNT
7551  *
7552          MOVE 1 TO WS-WHAT-SUB
7553          MOVE 0 TO WS-ASSM-COUNTER
7554          PERFORM UNTIL WS-ASSM-COUNTER = WS-COUNT-ASSM
7555          MOVE SPACES TO PRINT-LINE
7556          IF WS-ASDC-GB-DESC(WS-WHAT-SUB) = "KY MUNICIPAL TAX" THEN

```

```

7557      MOVE WS-CYCY-CDE-KY(WS-WHAT-SUB) TO WS-CYCY-CODE
7558      EXEC SQL AT ORA1
7559      SELECT UPPER(CYCY_GB_NAME)
7560      INTO :WS-CYCY-NAME
7561      FROM CYCY
7562      WHERE CYCY_GB_CODE = :WS-CYCY-CODE
7563      END-EXEC
7564      MOVE WS-ASDC-GB-DESC(WS-WHAT-SUB)
7565      TO PRT-KY-ASSM-COV-DESC
7566      MOVE " - " TO PRT-ASSM-DASH
7567      MOVE WS-CYCY-NAME
7568      TO PRT-KY-CITY-CNTY
7569      MOVE "(INCLUDES COLLECTION FEE)"
7570      TO PRT-KY-ASSM-FEE
7571      IF WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB) > 0 THEN
7572          MOVE WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB)
7573          TO PRT-KY-ASSM-COST-DECIMAL
7574      ELSE
7575          MOVE WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB)
7576          TO PRT-KY-ZERO-ASSM-COST-DECIMAL
7577      END-IF
7578      MOVE SPACES TO WS-CYCY-CODE
7579  ELSE
7580      MOVE WS-ASDC-GB-DESC(WS-WHAT-SUB)
7581      TO PRT-2A-ASSM-COV-DESC
7582      IF WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB) > 0 THEN
7583          MOVE WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB)
7584          TO PRT-ASSM-COST-DECIMAL
7585      ELSE
7586          MOVE WS-ASSM-GB-WRIT-AMT(WS-WHAT-SUB)
7587          TO PRT-ZERO-ASSM-COST-DECIMAL
7588      END-IF
7589  END-IF
7590  WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7591  ADD 1 TO WS-ASSM-COUNTER
7592  ADD 1 TO WS-WHAT-SUB
7593  ADD 1 TO WS-PG2-LINECNT
7594  END-PERFORM
7595  END-IF.
7596  *
7597  IF WS-SUM-ASSM-MVFEE-1403 = 'N' THEN
7598      MOVE SPACES TO PRINT-LINE
7599      MOVE "NY MOTOR VEHICLE LAW" TO PRT-2A-ASSM-COV-DESC
7600      MOVE WS-SUM-ASSM-MVFEE TO PRT-ZERO-ASSM-COST-DECIMAL
7601      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7602      ADD 1 TO WS-PG2-LINECNT
7603      MOVE SPACES TO PRINT-LINE
7604      MOVE " ENFORCEMENT FEE" TO PRT-2A-ASSM-COV-DESC
7605      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7606      ADD 1 TO WS-PG2-LINECNT
7607  END-IF.
7608  *
7609  IF WS-ASSM-GB-AMOUNT-POLY > 0 THEN
7610      MOVE SPACES TO PRINT-LINE
7611      MOVE WS-ASSM-GB-AMOUNT-POLY TO PRT-COST-DECIMAL
7612      WRITE PRINT-LINE AFTER ADVANCING 1 LINES

```

```

7613         ADD 1 TO WS-PG2-LINECNT
7614     END-IF.
7615 *
7616     MOVE 0 TO WS-ASSM-COUNTER.
7617     MOVE 0 TO WS-COUNT-ASSM.
7618     MOVE 0 TO WS-REDUCT.
7619     MOVE 'M' TO WS-MCTB-EDIT.
7620     PERFORM 2390-GET-POLICY-ASSM-INFO THRU 2390-EXIT.
7621     COMPUTE WS-ASSM-CNT = 41 - WS-COUNT-ASSM.
7622     IF WS-PG2-LINECNT >= WS-ASSM-CNT THEN
7623         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7624     END-IF
7625     MOVE 1 TO WS-WHAT-SUB.
7626     MOVE 0 TO WS-ASSM-COUNTER.
7627     PERFORM UNTIL WS-ASSM-COUNTER = WS-COUNT-ASSM
7628         MOVE SPACES TO PRINT-LINE
7629         MOVE WS-ASSM-REDUCT-DESC(WS-WHAT-SUB)
7630             TO PRT-COVERAGE-DESC
7631         IF WS-ASSM-REDUCT-WRIT-AMOUNT(WS-WHAT-SUB) > 0 THEN
7632             MOVE WS-ASSM-REDUCT-WRIT-AMOUNT(WS-WHAT-SUB)
7633                 TO PRT-COST-DECIMAL
7634         ELSE
7635             MOVE WS-ASSM-REDUCT-WRIT-AMOUNT(WS-WHAT-SUB)
7636                 TO PRT-ZERO-COST-DECIMAL
7637         END-IF
7638         WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7639         ADD 1 TO WS-ASSM-COUNTER
7640         ADD 1 TO WS-WHAT-SUB
7641         ADD 2 TO WS-PG2-LINECNT
7642     END-PERFORM.
7643 *
7644     MOVE SPACES TO PRINT-LINE.
7645     INITIALIZE WS-FOOTER.
7646     IF WS-PRT-2A = "Y" THEN
7647         SUBTRACT WS-PG2-LINECNT FROM 46 GIVING WS-FOOTER
7648     ELSE
7649         SUBTRACT WS-PG2-LINECNT FROM 41 GIVING WS-FOOTER
7650     END-IF.
7651     WRITE PRINT-LINE AFTER ADVANCING WS-FOOTER LINES.
7652     MOVE "ESTIMATED TOTAL PREMIUM" TO PRT-COVERAGE-DESC.
7653     INITIALIZE WS-PG2-LINECNT.
7654 *     MOVE 0 TO WS-PG2-LINECNT.
7655     MOVE WS-INFC-GB-ANNUAL-PX TO PRT-COST-DECIMAL.
7656     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
7657 *
7658     7225-EXIT.
7659     EXIT.
7660 *
7661 *     ADD TO PRINT LINE - RETURN THEN SKIP LINES
7662     7226-PATCH-WA.
7663         MOVE WS-WA-HOLD-LIMIT TO PRT-COVERAGE-LIMIT-A
7664         MOVE WS-WA-HOLD-PX TO PRT-COVERAGE-COST
7665         WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7666         ADD 2 TO WS-PG2-LINECNT.
7667     7226-EXIT.
7668     EXIT.

```



```

7669 *
7670 7230-PRT-ADD-PIP.
7671 *
7672 IF WS-NY-EXIST = "N" OR WS-MULTISTATE = "Y" THEN
7673 IF WS-PG2-LINECNT >= 36 THEN
7674 PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7675 END-IF
7676 MOVE SPACES TO PRINT-LINE
7677 MOVE "ADDED PERSONAL INJURY" TO PRT-COVERAGE-DESC
7678 MOVE "SEPARATELY STATED IN EACH"
7679 TO PRT-COVERAGE-SEP-STATED
7680 MOVE WS-INFC-AU-COV-AU-ADDED-PIP(1:5)
7681 TO PRT-COVERAGE-AUTO
7682 *****
7683 IF WS-VAR-ADPIP-EXIST = 'Y' THEN
7684 DISPLAY 'WS-VAR-ADPIP-EXIST = Y'
7685 IF WS-SUM-WRIT-PREM-ADDED-PIP not equal 0
7686 MOVE WS-SUM-WRIT-PREM-ADDED-PIP
7687 TO PRT-ZERO-COST-DECIMAL
7688 ELSE
7689 MOVE WS-SUM-WRIT-PREM-ADDED-PIP
7690 TO PRT-COST-DECIMAL
7691 end-if
7692 ELSE
7693 DISPLAY 'WS-VAR-ADPIP-EXIST = N'
7694 MOVE WS-SUM-WRIT-PREM-ADDED-PIP TO PRT-COST-DECIMAL
7695 END-IF
7696 WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7697 ADD 2 TO WS-PG2-LINECNT
7698 *
7699 MOVE SPACES TO PRINT-LINE
7700 MOVE " PROTECTION (OR EQUIVALENT" TO PRT-COVERAGE-DESC
7701 MOVE WS-INFC-AU-COV-AU-ADDED-PIP(7:3)
7702 TO PRT-COVERAGE-AUTO
7703 MOVE "ADDED PIP ENDORSEMENT"
7704 TO PRT-COVERAGE-SEP-STATED
7705 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7706 ADD 1 TO WS-PG2-LINECNT
7707 *
7708 MOVE SPACES TO PRINT-LINE
7709 MOVE " NO-FAULT COVERAGE)" TO PRT-COVERAGE-DESC
7710 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7711 ADD 1 TO WS-PG2-LINECNT
7712 END-IF.
7713 *
7714 IF WS-NY-EXIST = "Y" THEN
7715 IF WS-PG2-LINECNT >= 36 THEN
7716 PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7717 END-IF
7718 IF WS-MULTISTATE = "Y" THEN
7719 MOVE SPACES TO PRINT-LINE
7720 MOVE " NY ONLY:" TO PRT-COVERAGE-DESC-SM
7721 WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7722 ADD 2 TO WS-PG2-LINECNT
7723 END-IF
7724 *

```

```

7725 MOVE SPACES TO PRINT-LINE
7726 IF WS-MULTISTATE = "Y" THEN
7727     MOVE " MANDATORY BASIC ECONOMIC LOSS"
7728     TO PRT-COVERAGE-DESC-SM
7729 ELSE
7730     MOVE " MANDATORY BASIC ECONOMIC LOSS"
7731     TO PRT-COVERAGE-DESC-SM
7732 END-IF
7733 MOVE WS-NY-PIP-BASIC-LIMIT(1) TO PRT-COVERAGE-LIMIT-SM
7734 IF WS-NY-PIP-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
7735     MOVE WS-NY-WRIT-PREM-PIP TO PRT-ZERO-SUB-AUCV-COST-SM
7736 ELSE
7737     MOVE WS-NY-WRIT-PREM-PIP TO PRT-SUB-AUCV-COST-SM
7738 END-IF
7739 IF WS-MULTISTATE = "Y" THEN
7740     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7741     ADD 1 TO WS-PG2-LINECNT
7742 ELSE
7743     WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7744     ADD 2 TO WS-PG2-LINECNT
7745 END-IF
7746 *
7747     IF WS-PG2-LINECNT >= 40 THEN
7748         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7749     END-IF
7750 MOVE SPACES TO PRINT-LINE
7751 IF WS-MULTISTATE = "Y" THEN
7752     MOVE " OPTIONAL BASIC ECONOMIC LOSS"
7753     TO PRT-COVERAGE-DESC-SM
7754 ELSE
7755     MOVE " OPTIONAL BASIC ECONOMIC LOSS"
7756     TO PRT-COVERAGE-DESC-SM
7757 END-IF
7758 MOVE WS-OBEL-LIMIT(1) TO PRT-COVERAGE-LIMIT-SM
7759 IF WS-OBEL-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
7760     MOVE WS-SUM-WRIT-PREM-OBEL
7761     TO PRT-ZERO-SUB-AUCV-COST-SM
7762 ELSE
7763     MOVE WS-SUM-WRIT-PREM-OBEL TO PRT-SUB-AUCV-COST-SM
7764 END-IF
7765 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7766 ADD 1 TO WS-PG2-LINECNT
7767 *
7768 IF WS-PG2-LINECNT >= 39 THEN
7769     PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7770 END-IF
7771 MOVE SPACES TO PRINT-LINE
7772 IF WS-MULTISTATE = "Y" THEN
7773     MOVE " ADDITIONAL PERSONAL"
7774     TO PRT-COVERAGE-DESC-SM
7775 ELSE
7776     MOVE " ADDITIONAL PERSONAL"
7777     TO PRT-COVERAGE-DESC-SM
7778 END-IF
7779 MOVE WS-NY-APIP-BASIC-LIMIT(1)
7780 TO PRT-COVERAGE-LIMIT-SM

```

```

7781 IF WS-ADPIP-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
7782     MOVE WS-SUM-WRIT-PREM-NY-ADDED-PIP
7783     TO PRT-ZERO-SUB-AUCV-COST-SM
7784 ELSE
7785     MOVE WS-SUM-WRIT-PREM-NY-ADDED-PIP
7786     TO PRT-SUB-AUCV-COST-SM
7787 END-IF
7788 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7789 ADD 1 TO WS-PG2-LINECNT
7790 *
7791 MOVE SPACES TO PRINT-LINE
7792 IF WS-MULTISTATE = "Y" THEN
7793     MOVE " INJURY PROTECTION" TO PRT-COVERAGE-DESC-SM
7794 ELSE
7795     MOVE " INJURY PROTECTION" TO PRT-COVERAGE-DESC-SM
7796 END-IF
7797 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7798 ADD 1 TO WS-PG2-LINECNT
7799 *
7800 IF WS-NY-PIP-BASIC-LIMIT(1) = SPACES
7801     MOVE 0 TO WS-PIP-BASIC-LIMIT-NUM
7802 ELSE
7803     MOVE WS-NY-PIP-BASIC-LIMIT(1)
7804     TO WS-PIP-BASIC-LIMIT-NUM
7805 END-IF
7806 IF WS-NY-APIP-BASIC-LIMIT(1) = SPACES
7807     MOVE 0 TO WS-APIP-BASIC-LIMIT-NUM
7808 ELSE
7809     MOVE WS-NY-APIP-BASIC-LIMIT(1)
7810     TO WS-APIP-BASIC-LIMIT-NUM
7811 END-IF
7812 IF WS-OBEL-LIMIT(1) = SPACES
7813     MOVE 0 TO WS-OBEL-LIMIT-NUM
7814 ELSE
7815     MOVE WS-OBEL-LIMIT(1) TO WS-OBEL-LIMIT-NUM
7816 END-IF
7817 *
7818 IF WS-PG2-LINECNT >= 39 THEN
7819     PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7820 END-IF
7821 MOVE SPACES TO PRINT-LINE
7822 IF WS-MULTISTATE = "Y" THEN
7823     MOVE " AGGREGATE NO-FAULT " TO PRT-AGGR-DESC
7824 ELSE
7825     MOVE " AGGREGATE NO-FAULT " TO PRT-AGGR-DESC
7826 END-IF
7827 COMPUTE WS-SUM-TOTAL-PIP = WS-NY-WRIT-PREM-PIP
7828     + WS-SUM-WRIT-PREM-NY-ADDED-PIP
7829     + WS-SUM-WRIT-PREM-OBEL
7830 IF WS-NY-PIP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
7831     WS-ADPIP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
7832     WS-OBEL-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
7833     MOVE WS-SUM-TOTAL-PIP TO PRT-ZERO-SUM-TOTAL-PIP
7834 ELSE
7835     MOVE WS-SUM-TOTAL-PIP TO PRT-SUM-TOTAL-PIP
7836 END-IF

```

```

7837      COMPUTE WS-SUM-TL-PIP-LT = WS-PIP-BASIC-LIMIT-NUM
7838          + WS-APIP-BASIC-LIMIT-NUM
7839          + WS-OBEL-LIMIT-NUM
7840      MOVE WS-SUM-TL-PIP-LT TO PRT-SUM-TL-PIP-LT
7841      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7842      ADD 1 TO WS-PG2-LINECNT
7843  *
7844      MOVE SPACES          TO PRINT-LINE
7845      IF WS-MULTISTATE = "Y" THEN
7846          MOVE " BENEFITS AVAILABLE" TO PRT-AGGR-DESC
7847      ELSE
7848          MOVE " BENEFITS AVAILABLE" TO PRT-AGGR-DESC
7849      END-IF
7850      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7851      ADD 1 TO WS-PG2-LINECNT
7852  *
7853      IF WS-NYAPIP-EXIST = 'Y' and WS-VAR-NY-PIP-EXIST = 'N'
7854      THEN
7855          EVALUATE WS-NY-APIP-BASIC-LIMIT(1)
7856              WHEN "0025000"
7857                  MOVE "0002500" TO WS-PIP-WORK-LIMIT(1)
7858              WHEN "0050000"
7859                  MOVE "0003000" TO WS-PIP-WORK-LIMIT(1)
7860              WHEN "0100000"
7861                  MOVE "0004000" TO WS-PIP-WORK-LIMIT(1)
7862              WHEN "0150000"
7863                  MOVE "0005000" TO WS-PIP-WORK-LIMIT(1)
7864          END-EVALUATE
7865          MOVE "0000050" TO WS-PIP-EXP-LIMIT(1)
7866          MOVE "0002000" TO WS-PIP-DEATH-LIMIT(1)
7867      ELSE
7868          IF WS-VAR-NY-PIP-EXIST = 'Y' and
7869              WS-NYAPDTH-EXIST = 'N' THEN
7870              MOVE "0002000" TO WS-PIP-WORK-LIMIT(1)
7871              MOVE "0000025" TO WS-PIP-EXP-LIMIT(1)
7872              MOVE "0002000" TO WS-PIP-DEATH-LIMIT(1)
7873          END-IF
7874          IF WS-NYAPDTH-EXIST = 'Y' THEN
7875              MOVE "0003000" TO WS-PIP-WORK-LIMIT(1)
7876              MOVE "0000050" TO WS-PIP-EXP-LIMIT(1)
7877
7878              IF WS-NY-APDTH-ADD-LIMIT(1) = '0003000' THEN
7879                  MOVE "0005000" TO WS-PIP-DEATH-LIMIT(1)
7880              ELSE
7881                  MOVE "0002000" TO WS-PIP-DEATH-LIMIT(1)
7882              END-IF
7883
7884          END-IF
7885      END-IF
7886  *
7887      IF WS-PG2-LINECNT >= 39 THEN
7888          PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7889      END-IF
7890      MOVE SPACES TO PRINT-LINE
7891      IF WS-MULTISTATE = "Y" THEN
7892          MOVE " MAXIMUM MONTHLY WORK LOSS"

```

```

7893         TO PRT-COVERAGE-DESC-SM
7894     ELSE
7895         MOVE "    MAXIMUM MONTHLY WORK LOSS"
7896         TO PRT-COVERAGE-DESC-SM
7897     END-IF
7898     MOVE WS-PIP-WORK-LIMIT(1) TO PRT-COVERAGE-LIMIT-SM
7899     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7900     ADD 1 TO WS-PG2-LINECNT
7901 *
7902     IF WS-PG2-LINECNT >= 38 THEN
7903         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7904     END-IF
7905     MOVE SPACES TO PRINT-LINE
7906     IF WS-MULTISTATE = "Y" THEN
7907         MOVE "    OTHER NECESSARY EXPENSES"
7908         TO PRT-COVERAGE-DESC-SM
7909     ELSE
7910         MOVE "    OTHER NECESSARY EXPENSES"
7911         TO PRT-COVERAGE-DESC-SM
7912     END-IF
7913     MOVE WS-PIP-EXP-LIMIT(1) TO PRT-COVERAGE-LIMIT-SM
7914     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7915     ADD 1 TO WS-PG2-LINECNT
7916     MOVE SPACES TO PRINT-LINE
7917     IF WS-MULTISTATE = "Y" THEN
7918         MOVE "    (PER DAY)" TO PRT-COVERAGE-DESC-SM
7919     ELSE
7920         MOVE "    (PER DAY)" TO PRT-COVERAGE-DESC-SM
7921     END-IF
7922     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7923     ADD 1 TO WS-PG2-LINECNT
7924 *
7925     IF WS-PG2-LINECNT >= 39 THEN
7926         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7927     END-IF
7928     MOVE SPACES TO PRINT-LINE
7929     IF WS-MULTISTATE = "Y" THEN
7930         MOVE "    DEATH BENEFIT" TO PRT-COVERAGE-DESC-SM
7931     ELSE
7932         MOVE "    DEATH BENEFIT" TO PRT-COVERAGE-DESC-SM
7933     END-IF
7934     MOVE WS-PIP-DEATH-LIMIT(1) TO PRT-COVERAGE-LIMIT-SM
7935     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7936     ADD 1 TO WS-PG2-LINECNT
7937     END-IF.
7938 *
7939     IF WS-MI-EXIST = "Y" THEN
7940         IF WS-PG2-LINECNT >= 38 THEN
7941             PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
7942         END-IF
7943         MOVE SPACES          TO PRINT-LINE
7944         MOVE "PROPERTY PROTECTION"  TO PRT-COVERAGE-DESC
7945         MOVE "SEPARATELY STATED IN EACH"
7946         TO PRT-COVERAGE-SEP-STATED
7947         MOVE WS-INFC-AU-COV-AU-PPI(1:5) TO PRT-COVERAGE-AUTO
7948 *****

```

```

7949      IF WS-VAR-PPI-EXIST = 'Y' THEN
7950          MOVE WS-SUM-WRIT-PREM-PPI TO PRT-ZERO-COST-DECIMAL
7951      ELSE
7952          MOVE WS-SUM-WRIT-PREM-PPI TO PRT-COST-DECIMAL
7953      END-IF
7954      WRITE PRINT-LINE AFTER ADVANCING 2 LINES
7955      ADD 2 TO WS-PG2-LINECNT
7956  *
7957      MOVE SPACES          TO PRINT-LINE
7958      IF WS-MULTISTATE = "Y" THEN
7959          MOVE " INSURANCE (MI ONLY)" TO PRT-COVERAGE-DESC
7960      ELSE
7961          MOVE " INSURANCE " TO PRT-COVERAGE-DESC
7962      END-IF
7963      MOVE WS-INFC-AU-COV-AU-PPI(7:3) TO PRT-COVERAGE-AUTO
7964      MOVE "PPI ENDORSEMENT" TO PRT-COVERAGE-SEP-STATD
7965      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7966      ADD 1 TO WS-PG2-LINECNT
7967  **
7968      EXEC SQL AT ORA1
7969      SELECT EXIT_GB_DEDUCTIBLE
7970      INTO WS-PPI-DEDUCT
7971      FROM AUTO_COVERAGE, VEHICLE_CODES,
7972      VEHICLE_DESC, EXPOSURE, COIT
7973      WHERE EXIT_VHCD_AU_NO = VHCD_AU_NO
7974      AND EXIT_AUCV_AU_NO = AUCV_AU_NO
7975      AND EXIT_GB_NO = COIT_EXIT_GB_NO
7976      AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
7977      AND EXIT_VHDS_AU_NO = VHDS_AU_NO
7978      AND EXIT_VHDS_AU_NO = :WS-EXIT-VHDS-AU-NO
7979      AND EXIT_WC_CLASS_CODE IN ('PPI')
7980      END-EXEC
7981
7982      DISPLAY 'MICHIGAN EXISTS 3a: ' WS-MI-EXIST ' ' WS-PPI-DEDUCT
7983      DISPLAY 'MICHIGAN EXISTS 3b: ' WS-PPRT-INFC-GB-NO ' ' WS-EXIT-VHDS-AU-NO
7984
7985      IF WS-MI-EXIST = "Y" THEN
7986          if WS-PPI-DEDUCT > 0 then
7987              MOVE SPACES TO PRT-COVERAGE-DESC
7988              MOVE "$" TO PRT-PIP-COV-DESC
7989              MOVE WS-PPI-DEDUCT TO PRT-PIP-DEDUCT
7990              MOVE " DED (See Endorsement) " TO PRT-PIP-DESC
7991              WRITE PRINT-LINE AFTER ADVANCING 1 LINES
7992          end-if
7993      END-IF
7994  **
7995  *
7996      END-IF.
7997  *
7998      7230-EXIT.
7999      EXIT.
8000  *
8001      7235-PRT-STATE.
8002  *
8003      IF WS-MI-EXIST = "Y" THEN
8004          IF WS-PG2-LINECNT >= 38 THEN

```



```

8005         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
8006     END-IF
8007     MOVE SPACES                TO PRINT-LINE
8008     MOVE "PROPERTY DAMAGE LIABILITY" TO PRT-COVERAGE-DESC
8009     MOVE WS-INFC-AU-COV-AU-BUYBACK(1:5) TO PRT-COVERAGE-AUTO
8010 *****
8011     IF WS-VAR-BUY-EXIST = 'Y' THEN
8012         MOVE WS-SUM-WRIT-PREM-BUYBACK
8013             TO PRT-ZERO-COST-DECIMAL
8014     ELSE
8015         MOVE WS-SUM-WRIT-PREM-BUYBACK TO PRT-COST-DECIMAL
8016     END-IF
8017     WRITE PRINT-LINE AFTER ADVANCING 2 LINES
8018     ADD 2 TO WS-PG2-LINECNT
8019     MOVE SPACES                TO PRINT-LINE
8020     IF WS-MULTISTATE = "Y" THEN
8021         MOVE " COVERAGE BUYBACK (MI ONLY)"
8022             TO PRT-COVERAGE-DESC
8023     ELSE
8024         MOVE " COVERAGE BUYBACK " TO PRT-COVERAGE-DESC
8025     END-IF
8026     MOVE WS-INFC-AU-COV-AU-BUYBACK(7:3)
8027         TO PRT-COVERAGE-AUTO
8028     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
8029     ADD 1 TO WS-PG2-LINECNT
8030 END-IF.
8031 *
8032 *HIRED PHYSICAL DAMAGE
8033 *
8034     IF WS-SUM-WRIT-PREM-HIRPHY-1403 = 'N'
8035     IF WS-PG2-LINECNT >= 39 THEN
8036         PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
8037     END-IF
8038     MOVE SPACES TO PRINT-LINE
8039     MOVE "HIRED PHYSICAL DAMAGE" TO PRT-COVERAGE-DESC
8040     MOVE WS-SUM-WRIT-PREM-HIRPHY TO PRT-ZERO-COST-DECIMAL
8041     WRITE PRINT-LINE AFTER ADVANCING 2 LINES
8042     ADD 2 TO WS-PG2-LINECNT
8043 END-IF.
8044 *
8045 7235-EXIT.
8046 EXIT.
8047 *
8048 *-----*
8049 * THIS PARAGRAPH WILL PRINT A NON-SIMPLIFIED PAGE 2. *
8050 *-----*
8051 *
8052 7250-PRINT-NON-SIMPLIFIED.
8053 *
8054     PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8055     VARYING WS-BLANK-LINE FROM 1 BY 1
8056     UNTIL WS-BLANK-LINE > 25.
8057 *
8058     MOVE SPACES    TO PRINT-LINE.
8059     MOVE ESC TO FEED-ESC-6.
8060     WRITE PRINT-LINE FROM LINEFEED-6.

```

```

8061      *
8062      PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8063      VARYING WS-BLANK-LINE FROM 1 BY 1
8064      UNTIL WS-BLANK-LINE > 4.
8065      *
8066      MOVE SPACES TO PRINT-LINE.
8067      IF WS-INFC-AU-COV-AU-LIAB NOT EQUAL SPACES
8068      OR WS-SUM-WRIT-PREM-LIAB NOT EQUAL 0
8069      MOVE WS-INFC-AU-COV-AU-LIAB(1:5)
8070      TO PRT-COVERAGE-AUTO
8071      IF WS-LIAB-LIMIT(1) = "SEE END"
8072      AND WS-LIAB-LIMIT(2) = "0000000"
8073      MOVE " (SEE ENDORSEMENT)"
8074      TO PRT-COVERAGE-SEP-STATED
8075      ELSE
8076      IF WS-LIAB-LIMIT(1) NUMERIC
8077      MOVE WS-LIAB-LIMIT(1) TO PRT-COVERAGE-LIMIT
8078      ELSE
8079      MOVE WS-LIAB-LIMIT(1)
8080      TO PRT-COVERAGE-LIMIT-CHAR
8081      END-IF
8082      MOVE " EACH ACCIDENT" TO PRT-COVERAGE-TEXT
8083      END-IF
8084      MOVE WS-SUM-WRIT-PREM-LIAB TO PRT-COST-DECIMAL
8085      END-IF.
8086      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8087      ADD 1 TO WS-PG2-LINECNT.
8088      *
8089      MOVE SPACES TO PRINT-LINE.
8090      MOVE WS-INFC-AU-COV-AU-LIAB(7:3) TO PRT-COVERAGE-AUTO.
8091      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8092      ADD 1 TO WS-PG2-LINECNT.
8093      *
8094      MOVE SPACES TO PRINT-LINE.
8095      MOVE WS-INFC-AU-COV-AU-PIP(1:5) TO PRT-COVERAGE-AUTO.
8096      *****
8097      IF WS-VAR-PIP-EXIST = 'Y' THEN
8098      MOVE WS-SUM-WRIT-PREM-PIP TO PRT-ZERO-COST-DECIMAL
8099      ELSE
8100      MOVE WS-SUM-WRIT-PREM-PIP TO PRT-COST-DECIMAL
8101      END-IF.
8102      WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
8103      ADD 2 TO WS-PG2-LINECNT.
8104      *
8105      MOVE SPACES TO PRINT-LINE.
8106      MOVE WS-INFC-AU-COV-AU-PIP(7:3) TO PRT-COVERAGE-AUTO.
8107      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8108      ADD 1 TO WS-PG2-LINECNT.
8109      *
8110      MOVE SPACES TO PRINT-LINE.
8111      WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
8112      ADD 2 TO WS-PG2-LINECNT.
8113      *
8114      MOVE SPACES TO PRINT-LINE.
8115      MOVE ESC TO FEED-ESC-6.
8116      WRITE PRINT-LINE FROM LINEFEED-6.

```

```

8117 *
8118     MOVE SPACES                TO PRINT-LINE.
8119     MOVE WS-INFC-AU-COV-AU-ADDED-PIP(1:5) TO PRT-COVERAGE-AUTO.
8120 *****
8121     IF WS-VAR-ADPIP-EXIST = 'Y' THEN
8122         MOVE WS-SUM-WRIT-PREM-ADDED-PIP TO PRT-ZERO-COST-DECIMAL
8123     ELSE
8124         MOVE WS-SUM-WRIT-PREM-ADDED-PIP TO PRT-COST-DECIMAL
8125     END-IF.
8126     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8127     ADD 1 TO WS-PG2-LINECNT.
8128 *
8129     MOVE SPACES                TO PRINT-LINE.
8130     MOVE WS-INFC-AU-COV-AU-ADDED-PIP(7:3) TO PRT-COVERAGE-AUTO.
8131     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8132     ADD 1 TO WS-PG2-LINECNT.
8133 *
8134     MOVE SPACES TO PRINT-LINE.
8135     IF WS-INFC-AU-COV-AU-MED-PAY NOT EQUAL SPACES
8136        OR WS-SUM-WRIT-PREM-MED-PAY NOT EQUAL 0
8137        MOVE WS-INFC-AU-COV-AU-MED-PAY(1:5)
8138        TO PRT-COVERAGE-AUTO
8139        IF WS-MED-PAY-LIMIT(1) = "SEE END"
8140            AND WS-MED-PAY-LIMIT(2) = "0000000"
8141            MOVE " (SEE ENDORSEMENT)"
8142            TO PRT-COVERAGE-SEP-STATED
8143        ELSE
8144            IF WS-MED-PAY-LIMIT(1) NUMERIC
8145                MOVE WS-MED-PAY-LIMIT(1)
8146                TO PRT-COVERAGE-LIMIT
8147            ELSE
8148                MOVE WS-MED-PAY-LIMIT(1)
8149                TO PRT-COVERAGE-LIMIT-CHAR
8150        END-IF
8151    END-IF
8152    MOVE WS-SUM-WRIT-PREM-MED-PAY TO PRT-COST-DECIMAL
8153    END-IF.
8154    WRITE PRINT-LINE AFTER ADVANCING 4 LINES.
8155    ADD 4 TO WS-PG2-LINECNT.
8156 *
8157    MOVE SPACES                TO PRINT-LINE.
8158    MOVE WS-INFC-AU-COV-AU-MED-PAY(7:3) TO PRT-COVERAGE-AUTO.
8159    WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8160    ADD 1 TO WS-PG2-LINECNT.
8161 *
8162    MOVE SPACES TO PRINT-LINE.
8163    IF WS-INFC-AU-COV-AU-UM NOT EQUAL SPACES
8164        OR WS-SUM-WRIT-PREM-UM NOT EQUAL 0
8165        MOVE WS-INFC-AU-COV-AU-UM(1:5) TO PRT-COVERAGE-AUTO
8166        IF WS-UM-LIMIT(1) = "SEE END"
8167            AND WS-UM-LIMIT(2) = "0000000"
8168            MOVE " (SEE ENDORSEMENT)"
8169            TO PRT-COVERAGE-SEP-STATED
8170        ELSE
8171            IF WS-UM-LIMIT(1) NUMERIC
8172                MOVE WS-UM-LIMIT(1) TO PRT-COVERAGE-LIMIT

```

```

8173         ELSE
8174             MOVE WS-UM-LIMIT(1)
8175             TO PRT-COVERAGE-LIMIT-CHAR
8176         END-IF
8177     END-IF
8178     MOVE WS-SUM-WRIT-PREM-UM TO PRT-COST-DECIMAL
8179 END-IF.
8180 WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
8181 ADD 2 TO WS-PG2-LINECNT.
8182 *
8183     MOVE SPACES             TO PRINT-LINE.
8184     MOVE WS-INFC-AU-COV-AU-UM(7:3) TO PRT-COVERAGE-AUTO.
8185     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8186     ADD 1 TO WS-PG2-LINECNT.
8187 *
8188     MOVE SPACES             TO PRINT-LINE.
8189     MOVE WS-INFC-AU-COV-AU-COMP(1:5) TO PRT-COVERAGE-AUTO.
8190     IF WS-INFC-AU-COV-AU-COMP(1:5) NOT EQUAL SPACES THEN
8191         MOVE WS-SUM-WRIT-PREM-COMP TO PRT-ZERO-COST-DECIMAL
8192     ELSE
8193         MOVE WS-SUM-WRIT-PREM-COMP TO PRT-COST-DECIMAL
8194     END-IF.
8195     WRITE PRINT-LINE AFTER ADVANCING 4 LINES.
8196     ADD 4 TO WS-PG2-LINECNT.
8197 *
8198     MOVE SPACES             TO PRINT-LINE.
8199     MOVE WS-INFC-AU-COV-AU-COMP(7:3) TO PRT-COVERAGE-AUTO.
8200     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8201     ADD 1 TO WS-PG2-LINECNT.
8202 *
8203     MOVE SPACES             TO PRINT-LINE.
8204     MOVE WS-INFC-AU-COV-AU-SPECIFIED(1:5) TO PRT-COVERAGE-AUTO.
8205 *****
8206     IF WS-VAR-SPEC-LOSS-EXIST = 'Y' THEN
8207         MOVE WS-SUM-WRIT-PREM-SPEC TO PRT-ZERO-COST-DECIMAL
8208     ELSE
8209         MOVE WS-SUM-WRIT-PREM-SPEC TO PRT-COST-DECIMAL
8210     END-IF.
8211     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
8212     ADD 2 TO WS-PG2-LINECNT.
8213 *
8214     MOVE SPACES             TO PRINT-LINE.
8215     MOVE WS-INFC-AU-COV-AU-SPECIFIED(7:3) TO PRT-COVERAGE-AUTO.
8216     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8217     ADD 1 TO WS-PG2-LINECNT.
8218 *
8219     MOVE SPACES             TO PRINT-LINE.
8220     MOVE WS-INFC-AU-COV-AU-COLLISION(1:5) TO PRT-COVERAGE-AUTO.
8221     IF WS-INFC-AU-COV-AU-COLLISION (1:5) NOT EQUAL SPACES THEN
8222         MOVE WS-SUM-WRIT-PREM-COLL TO PRT-ZERO-COST-DECIMAL
8223     ELSE
8224         MOVE WS-SUM-WRIT-PREM-COLL TO PRT-COST-DECIMAL
8225     END-IF.
8226     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
8227     ADD 2 TO WS-PG2-LINECNT.
8228 *

```

```

8229         MOVE SPACES                TO PRINT-LINE.
8230         MOVE WS-INFC-AU-COV-AU-COLLISION(7:3) TO PRT-COVERAGE-AUTO.
8231         WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8232         ADD 1 TO WS-PG2-LINECNT.
8233     *
8234         MOVE SPACES                TO PRINT-LINE.
8235         MOVE WS-INFC-AU-COV-AU-TOW-LABOR(1:5) TO PRT-COVERAGE-AUTO.
8236     *****
8237         IF WS-VAR-TOW-EXIST = 'Y' THEN
8238             MOVE WS-SUM-WRIT-PREM-TOW TO PRT-ZERO-COST-DECIMAL
8239         ELSE
8240             MOVE WS-SUM-WRIT-PREM-TOW TO PRT-COST-DECIMAL
8241         END-IF.
8242         WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
8243         ADD 2 TO WS-PG2-LINECNT.
8244     *
8245         MOVE SPACES                TO PRINT-LINE.
8246         MOVE WS-INFC-AU-COV-AU-TOW-LABOR(7:3) TO PRT-COVERAGE-AUTO.
8247         WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8248         ADD 1 TO WS-PG2-LINECNT.
8249     *
8250         MOVE SPACES TO PRINT-LINE.
8251         IF WS-INFC-AU-COV-AU-RENTAL NOT EQUAL SPACES
8252             MOVE "RENTAL REIMBURSEMENT COV." TO PRT-COVERAGE-DESC
8253             MOVE WS-INFC-AU-COV-AU-RENTAL TO PRT-COVERAGE-AUTO
8254             MOVE WS-SUM-WRIT-PREM-RENTAL TO PRT-ZERO-COST-DECIMAL
8255         END-IF.
8256         WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8257         ADD 1 TO WS-PG2-LINECNT.
8258     *
8259         IF WS-INFC-AU-COV-AU-RENTAL(6:5) NOT EQUAL SPACES
8260             IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
8261                 DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
8262                 PERFORM 9999-DISPLAY-ERROR THRU 9999-EXIT
8263             ELSE
8264                 MOVE "**ERROR* UNABLE TO PRINT ALL AUTO SYMBOLS FOR RENTAL."
8265                     TO WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6)
8266                 ADD 1 TO WS-ENTRY-COUNT-6
8267             END-IF
8268         END-IF.
8269     *
8270     *HIRED PHYSICAL DAMAGE
8271     *
8272         MOVE SPACES TO PRINT-LINE.
8273         IF WS-SUM-WRIT-PREM-HIRPHY-1403 = 'N'
8274             MOVE "HIRED PHYSICAL DAMAGE" TO PRT-COVERAGE-DESC
8275             MOVE WS-SUM-WRIT-PREM-HIRPHY TO PRT-ZERO-COST-DECIMAL
8276         END-IF.
8277         WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8278         ADD 1 TO WS-PG2-LINECNT.
8279     *
8280         MOVE SPACES TO PRINT-LINE.
8281         IF WS-PRINT-PAGE-2A >= 2
8282         OR WS-CREDIT-COUNT >= 1
8283         THEN MOVE " (SEE EXTENDED SCHEDULE)"
8284             TO PRT-COVERAGE-SEP-STATED

```

```

8285         MOVE WS-ASSM-GB-AMOUNT TO PRT-COST-DECIMAL
8286         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
8287         ADD 1 TO WS-PG2-LINECNT
8288     ELSE     MOVE WS-ASSM-GB-AMOUNT TO PRT-COST-DECIMAL
8289             WRITE PRINT-LINE AFTER ADVANCING 1 LINES
8290             ADD 1 TO WS-PG2-LINECNT
8291     END-IF.
8292 *
8293     MOVE SPACES TO PRINT-LINE.
8294     MOVE WS-INFC-GB-ANNUAL-PX TO PRT-COST-DECIMAL.
8295     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8296     ADD 1 TO WS-PG2-LINECNT.
8297 *
8298     7250-EXIT.
8299     EXIT.
8300 *-----*
8301 * THIS PARAGRAPH WILL PRINT PAGE 2A OF THE POLICY, IF IT IS *
8302 * NECESSARY. *
8303 *-----*
8304 *
8305     7275-PRINT-PAGE-2A.
8306 *
8307     MOVE SPACES TO PRINT-LINE.
8308     INITIALIZE WS-FOOTER.
8309     IF WS-PRT-2A = "Y" THEN
8310         SUBTRACT WS-PG2-LINECNT FROM 46 GIVING WS-FOOTER
8311     ELSE
8312         SUBTRACT WS-PG2-LINECNT FROM 41 GIVING WS-FOOTER
8313     END-IF.
8314     WRITE PRINT-LINE AFTER ADVANCING WS-FOOTER LINES.
8315 *
8316     MOVE "Y" TO WS-PRT-2A.
8317 *
8318     MOVE SPACES TO PRINT-LINE.
8319     MOVE "***CONTINUED***" TO PRT-COVERAGE-SEP-STATED.
8320     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8321 *
8322     MOVE SPACES TO PRINT-LINE.
8323     MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-4-INFO.
8324     WRITE PRINT-LINE AFTER ADVANCING 3 LINES.
8325 *
8326     INITIALIZE WS-PG2-LINECNT.
8327 *
8328     MOVE 'AUCOV2' TO WS-PLFD-GB-FORM-DESC.
8329     MOVE 'Y' TO WS-PLFD-GB-ENABLE.
8330     MOVE 'NS' TO WS-PLFD-GB-OPTION-1.
8331     MOVE 'S' TO WS-PLFD-GB-OPTION-2.
8332 *
8333     IF WS-INFC-GB-PRIMARY-STATE NOT EQUAL SPACES THEN
8334         MOVE WS-INFC-GB-PRIMARY-STATE TO WS-PLFD-GB-STATE
8335     ELSE
8336         MOVE WS-CLIENT-STATE TO WS-PLFD-GB-STATE
8337     END-IF.
8338 *
8339     PERFORM 8500-GET-MACRO THRU 8500-EXIT.
8340 *

```



```

8341 IF SQLCODE EQUAL 1403
8342 MOVE 'XX' TO WS-PLFD-GB-STATE
8343 PERFORM 8500-GET-MACRO THRU 8500-EXIT
8344 END-IF.
8345 *
8346 IF SQLCODE NOT EQUAL 0
8347 MOVE "ERROR SELECTING MACRO FOR STATE XX - 7275"
8348 TO WS-ERR-MSG
8349 PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
8350 END-IF.
8351 *
8352 MOVE SPACES TO WS-PRINT-MACRO-LINE.
8353 MOVE ESC TO WS-PRINT-ESC.
8354 MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO.
8355 MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE.
8356 WRITE PRINT-LINE AFTER ADVANCING PAGE.
8357 *
8358 MOVE SPACES TO PRINT-LINE.
8359 MOVE ESC TO FEED-ESC-6.
8360 MOVE ESC TO COL-ESC.
8361 WRITE PRINT-LINE FROM LINEFEED-6.
8362 *
8363 PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8364 VARYING WS-BLANK-LINE FROM 1 BY 1
8365 UNTIL WS-BLANK-LINE > 7.
8366 *
8367 7275-EXIT.
8368 EXIT.
8369 *-----*
8370 * THIS PARAGRAPH WILL PRINT PAGE 3 ITEM FOUR, FIVE AND SIX OF THE*
8371 * AUTO POLICY. *
8372 *-----*
8373 *
8374 7300-PRINT-HIRED-NON-OWNED.
8375
8376
8377 INITIALIZE WS-VT-WI-LOC,
8378 WS-WI-IL-LOC,
8379 WS-VT-IL-LOC.
8380
8381
8382 *
8383 MOVE 'AUEND' TO WS-PLFD-GB-FORM-DESC.
8384 MOVE 'Y' TO WS-PLFD-GB-ENABLE.
8385 MOVE 'XX' TO WS-PLFD-GB-OPTION-1.
8386 MOVE 'XX' TO WS-PLFD-GB-OPTION-2.
8387 *
8388 IF WS-INFC-GB-PRIMARY-STATE NOT EQUAL SPACES THEN
8389 MOVE WS-INFC-GB-PRIMARY-STATE TO WS-PLFD-GB-STATE
8390 ELSE
8391 MOVE WS-CLIENT-STATE TO WS-PLFD-GB-STATE
8392 END-IF.
8393 *
8394 PERFORM 8500-GET-MACRO THRU 8500-EXIT.
8395 *
8396 IF SQLCODE EQUAL 1403

```

```

8397         MOVE 'XX' TO WS-PLFD-GB-STATE
8398         PERFORM 8500-GET-MACRO THRU 8500-EXIT
8399     END-IF.
8400 *
8401     IF SQLCODE NOT EQUAL 0
8402         MOVE "ERROR SELECTING MACRO FOR STATE XX - 7300"
8403         TO WS-ERR-MSG
8404         PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
8405     END-IF.
8406 *
8407     MOVE SPACES          TO WS-PRINT-MACRO-LINE.
8408     MOVE ESC             TO WS-PRINT-ESC.
8409     MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO.
8410     MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE.
8411     WRITE PRINT-LINE AFTER ADVANCING PAGE.
8412 *
8413     MOVE SPACES          TO PRINT-LINE.
8414     MOVE WS-COURIER       TO PRT-MISC-INFO.
8415     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
8416 *
8417     PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8418         VARYING WS-BLANK-LINE FROM 1 BY 1
8419         UNTIL WS-BLANK-LINE > 17.
8420 *
8421     MOVE SPACES TO PRINT-LINE.
8422     MOVE WS-SUM-WRIT-PREM-HIRED      TO PRT-HN-COST-DECIMAL.
8423     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
8424
8425 *
8426
8427     EXEC SQL AT ORA1
8428         SELECT COUNT(DISTINCT LOCT_GB_NO)
8429         INTO :WS-CNT-LA-LOC
8430         FROM COVERED_ITEM, LOC, INFC, POLY
8431         WHERE COIT_LOCT_GB_NO = LOCT_GB_NO
8432         AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
8433         AND INFC_GB_NO = COIT_INFC_GB_NO
8434         AND INFC_POLY_GB_NO_PART_OF = POLY_GB_NO
8435         AND POLY_GB_EFF_DATE > '31-DEC-2004'
8436         AND LOCT_STAT_GB_ALPHA_STATE_IN = 'LA'
8437     END-EXEC.
8438 *
8439     IF SQLCODE NOT EQUAL 0
8440     AND SQLCODE NOT EQUAL 1403
8441         MOVE "ERROR COUNTING LA LOCATIONS " TO WS-ERR-MSG
8442         PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
8443     END-IF.
8444 *
8445     IF SQLCODE EQUAL 1403 OR WS-CNT-LA-LOC = 0 THEN
8446         MOVE "N" TO WS-LA-LOC.
8447 *
8448     IF WS-CNT-LA-LOC > 0 THEN
8449         MOVE "Y" TO WS-LA-LOC.
8450
8451 *
8452     EXEC SQL AT ORA1

```

```

8453         SELECT COUNT(DISTINCT LOCT_GB_NO)
8454         INTO :WS-CNT-IL-LOC
8455         FROM COVERED_ITEM, LOC, INFC, POLY
8456         WHERE COIT_LOCT_GB_NO = LOCT_GB_NO
8457         AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
8458         AND INFC_GB_NO = COIT_INFC_GB_NO
8459         AND INFC_POLY_GB_NO_PART_OF = POLY_GB_NO
8460         AND POLY_GB_EFF_DATE > '31-DEC-2004'
8461         AND LOCT_STAT_GB_ALPHA_STATE_IN = 'IL'
8462     END-EXEC.
8463 *
8464     IF SQLCODE NOT EQUAL 0
8465     AND SQLCODE NOT EQUAL 1403
8466     MOVE "ERROR COUNTING IL LOCATIONS " TO WS-ERR-MSG
8467     PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
8468     END-IF.
8469 *
8470     IF SQLCODE EQUAL 1403 OR WS-CNT-IL-LOC = 0 THEN
8471     MOVE "N" TO WS-IL-LOC.
8472 *
8473     IF WS-CNT-IL-LOC > 0 THEN
8474     MOVE "Y" TO WS-IL-LOC.
8475 *
8476     EXEC SQL AT ORA1
8477     SELECT COUNT(DISTINCT LOCT_GB_NO)
8478     INTO :WS-CNT-WI-LOC
8479     FROM COVERED_ITEM, LOC, INFC, POLY
8480     WHERE COIT_LOCT_GB_NO = LOCT_GB_NO
8481     AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
8482     AND INFC_GB_NO = COIT_INFC_GB_NO
8483     AND INFC_POLY_GB_NO_PART_OF = POLY_GB_NO
8484     AND POLY_GB_EFF_DATE > '31-DEC-2007'
8485     AND LOCT_STAT_GB_ALPHA_STATE_IN = 'WI'
8486     END-EXEC.
8487 *
8488     IF SQLCODE NOT EQUAL 0
8489     AND SQLCODE NOT EQUAL 1403
8490     MOVE "ERROR COUNTING WI LOCATIONS " TO WS-ERR-MSG
8491     PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
8492     END-IF.
8493 *
8494     IF SQLCODE EQUAL 1403 OR WS-CNT-WI-LOC = 0 THEN
8495     MOVE "N" TO WS-WI-LOC.
8496 *****-- PLI32502.17526 A
8497 *
8498     EXEC SQL AT ORA1
8499     SELECT COUNT(DISTINCT LOCT_GB_NO)
8500     INTO :WS-CNT-VT-LOC
8501     FROM COVERED_ITEM, LOC, INFC, POLY
8502     WHERE COIT_LOCT_GB_NO = LOCT_GB_NO
8503     AND COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
8504     AND INFC_GB_NO = COIT_INFC_GB_NO
8505     AND INFC_POLY_GB_NO_PART_OF = POLY_GB_NO
8506     AND POLY_GB_EFF_DATE > '31-JAN-2017'
8507     AND LOCT_STAT_GB_ALPHA_STATE_IN = 'VT'
8508     END-EXEC.

```

```

8509      *
8510      IF SQLCODE NOT EQUAL 0
8511          AND SQLCODE NOT EQUAL 1403
8512          MOVE "ERROR COUNTING IN LOCATIONS " TO WS-ERR-MSG
8513          PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
8514      END-IF.
8515      *
8516      IF SQLCODE EQUAL 1403 OR WS-CNT-VT-LOC = 0 THEN
8517          MOVE "N" TO WS-VT-LOC.
8518      *****
8519      *
8520          IF WS-IL-LOC = 'Y'
8521              MOVE "Illinois" TO WS-STATE-VAR9
8522          END-IF.
8523
8524          IF WS-VT-LOC = 'Y'
8525              MOVE "Vermont" TO WS-STATE-VAR7
8526          END-IF.
8527
8528          IF WS-WI-LOC = 'Y'
8529              MOVE "Wisconsin" TO WS-STATE-VAR9
8530          END-IF.
8531
8532          IF WS-LA-LOC = 'Y'
8533              MOVE "Louisiana" TO WS-STATE-VAR9
8534          END-IF.
8535
8536
8537
8538          IF WS-IL-LOC = 'Y' AND WS-VT-LOC = 'Y'
8539              MOVE "Illinois or Vermont" TO WS-STATE-VAR22
8540              MOVE 'Y' TO WS-VT-IL-LOC
8541              MOVE 'N' TO WS-IL-LOC
8542              MOVE 'N' TO WS-VT-LOC
8543          END-IF.
8544
8545          IF WS-VT-LOC = 'Y' AND WS-WI-LOC = 'Y'
8546              MOVE "Vermont or Wisconsin" TO WS-STATE-VAR22
8547              MOVE "Y" TO WS-VT-WI-LOC
8548          END-IF.
8549
8550          IF WS-IL-LOC = 'Y' AND WS-WI-LOC = 'Y'
8551              MOVE "Wisconsin" TO WS-STATE-VAR9
8552              MOVE "Y" TO WS-WI-IL-LOC
8553          END-IF.
8554
8555          IF WS-VT-WI-LOC = 'Y'
8556              IF WS-VT-IL-LOC = 'Y'
8557                  MOVE "Wisconsin" TO WS-STATE-VAR9
8558                  MOVE SPACES TO WS-VT-WI-LOC
8559              END-IF
8560          END-IF.
8561
8562          IF WS-COVG-TEXT IS EQUAL SPACES
8563              IF WS-IL-LOC NOT EQUAL 'Y'
8564                  MOVE 'UNINSURED' TO WS-IL-COVGTEXT

```

```

8565      END-IF
8566      END-IF.
8567
8568
8569      IF  WS-IL-LOC  = 'N' AND
8570          WS-VT-LOC  = 'N' AND
8571          WS-WI-LOC  = 'N'
8572          MOVE 'N' TO WS-WI-IL-LOC
8573          MOVE 'N' TO WS-VT-IL-LOC
8574      END-IF.
8575
8576      *****--
8577      IF WS-MED-HOLD = 'Y'
8578          IF WS-HA-IND = 'Y'
8579              MOVE 'Y' TO WS-HAMED-IND
8580          END-IF
8581
8582          IF WS-NO-IND = 'Y'
8583              MOVE 'Y' TO WS-NOMED-IND
8584          END-IF
8585      END-IF.
8586      *****--
8587
8588      DISPLAY '*****'.
8589      DISPLAY 'POLICY: ' WS-POLY-NO.
8590      DISPLAY 'HA/NO: ' WS-HA-IND '/' WS-NO-IND.
8591      DISPLAY 'UI/UM: ' WS-AUCV-CODE.
8592      DISPLAY 'MEDICAL: ' WS-MED-HOLD.
8593      DISPLAY '7300 IL: ' WS-IL-LOC ' VT: ' WS-VT-LOC ' WI: ' WS-WI-LOC ' LA: ' WS-LA-LOC.
8594      DISPLAY 'IL/WI: ' WS-WI-IL-LOC ' IL/VT: ' WS-VT-IL-LOC.
8595      DISPLAY WS-STATE-VAR22 ' ' WS-IL-COVGTEXT.
8596      display 'LIABILITY?: ' WS-AUCV-AU-TYPE
8597      DISPLAY '*****'.
8598
8599
8600
8601
8602      *****
8603      *****-- THIS IS THE HIRED AUTO SECTION
8604
8605      IF WS-HA-IND = 'Y'
8606
8607          IF WS-VT-IL-LOC = "Y" OR WS-IL-LOC = "Y" THEN
8608              MOVE SPACES TO PRINT-LINE
8609              STRING WS-IL-COVGTEXT,
8610                  " Motorists Coverage - This coverage is afforded
8611      -      " for vehicle(s)"
8612              DELIMITED BY SIZE
8613              INTO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8614              ADD 1 TO WS-HIRED-AUTO-COUNT
8615      *
8616
8617          IF WS-VT-IL-LOC = 'Y'
8618              MOVE SPACES TO PRINT-LINE
8619              STRING "garaged in ",
8620                  WS-STATE-VAR22,
8621                  " when Hired Auto coverage and ",

```

```

8621         WS-IL-COVGTEXT
8622         DELIMITED BY SIZE
8623         INTO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8624         ADD 1 TO WS-HIRED-AUTO-COUNT
8625     ELSE
8626         IF WS-IL-LOC = 'Y'
8627             MOVE SPACES TO PRINT-LINE
8628             MOVE 'Illinois' TO WS-STATE-VAR9
8629             STRING "garaged in ",
8630             WS-STATE-VAR9,
8631             "(only) when Hired Auto coverage and ",
8632             WS-IL-COVGTEXT
8633             DELIMITED BY SIZE
8634             INTO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8635             ADD 1 TO WS-HIRED-AUTO-COUNT
8636         END-IF
8637     end-if
8638
8639     *
8640     MOVE SPACES TO PRINT-LINE
8641     MOVE "Motorists coverage are provided."
8642     TO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8643     ADD 1 TO WS-HIRED-AUTO-COUNT
8644     *
8645     MOVE SPACES TO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8646     ADD 1 TO WS-HIRED-AUTO-COUNT
8647
8648     if WS-IL-LOC = "Y"
8649
8650         MOVE SPACES TO PRINT-LINE
8651         MOVE WS-12-CHAR TO PRT-MISC-INFO
8652         WRITE PRINT-LINE
8653
8654
8655         PERFORM 73001-PRINT-UIUM-HIRED VARYING WS-HIRED-AUTO-COUNT
8656             FROM 1 BY 1 UNTIL WS-HIRED-AUTO-COUNT = 4
8657     end-if
8658
8659
8660
8661     *
8662     END-IF
8663     *
8664     IF WS-WI-LOC = "Y" THEN
8665
8666         MOVE SPACES TO PRINT-LINE
8667         MOVE WS-12-CHAR TO PRT-MISC-INFO
8668         WRITE PRINT-LINE
8669     *
8670     MOVE SPACES TO PRINT-LINE
8671     MOVE "Uninsured Motorists Coverage - This coverage is
8672     - " afforded for vehicle(s) garaged"
8673     TO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8674     ADD 1 TO WS-HIRED-AUTO-COUNT
8675     *
8676     MOVE SPACES TO PRINT-LINE

```



```

8677      MOVE SPACES TO PRT-IL-COVERAGE-DESC
8678      MOVE "in Wisconsin (only) when Hired Auto coverage and
8679      -    " Uninsured Motorists"
8680          TO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8681          ADD 1 TO WS-HIRED-AUTO-COUNT
8682      *
8683      MOVE SPACES TO PRINT-LINE
8684      MOVE "coverage are provided."
8685          TO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8686          ADD 1 TO WS-HIRED-AUTO-COUNT
8687      *
8688      MOVE SPACES TO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8689          ADD 1 TO WS-HIRED-AUTO-COUNT
8690      *****
8691      IF WS-HAMED-IND = 'Y'
8692          STRING "Medical Payments Coverage - This coverage
8693      -    " is afforded for vehicle(s) garaged in "
8694          DELIMITED BY SIZE
8695          INTO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8696          ADD 1 TO WS-HIRED-AUTO-COUNT
8697      *
8698          STRING "Wisconsin (only) when Hired Auto coverage
8699      -    "and Medical Payments coverage are"
8700          DELIMITED BY SIZE
8701          INTO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8702          ADD 1 TO WS-HIRED-AUTO-COUNT
8703      *
8704          STRING "provided."
8705          DELIMITED BY SIZE
8706          INTO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8707          ADD 1 TO WS-HIRED-AUTO-COUNT
8708      *
8709          MOVE SPACES TO WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
8710          ADD 1 TO WS-HIRED-AUTO-COUNT
8711      END-IF
8712      ADD 1 TO WS-HIRED-AUTO-COUNT
8713      *****
8714
8715      IF WS-HAMED-IND = 'Y'
8716          PERFORM 73001-PRINT-UIUM-HIRED VARYING WS-HIRED-AUTO-COUNT
8717              FROM 1 BY 1 UNTIL WS-HIRED-AUTO-COUNT > 7
8718
8719
8720          MOVE 'AUCOV3'    TO WS-PLFD-GB-FORM-DESC
8721          MOVE 'Y'         TO WS-PLFD-GB-ENABLE
8722          MOVE 'XX'        TO WS-PLFD-GB-OPTION-1
8723          MOVE 'XX'        TO WS-PLFD-GB-OPTION-2
8724      *
8725          MOVE 'XX' TO WS-PLFD-GB-STATE
8726          PERFORM 8500-GET-MACRO THRU 8500-EXIT
8727      else
8728      if WS-WI-LOC = "Y" THEN
8729          PERFORM 73001-PRINT-UIUM-HIRED VARYING WS-HIRED-AUTO-COUNT
8730              FROM 1 BY 1 UNTIL WS-HIRED-AUTO-COUNT = 8
8731      else
8732          PERFORM 73001-PRINT-UIUM-HIRED VARYING WS-HIRED-AUTO-COUNT

```

```

8733             FROM 1 BY 1 UNTIL WS-HIREDAUTO-COUNT = 6
8734         end-if
8735     end-if
8736
8737
8738     END-IF
8739
8740     *****
8741
8742     IF WS-VT-LOC = "Y" THEN
8743         MOVE SPACES   TO PRINT-LINE
8744         MOVE WS-12-CHAR TO PRT-MISC-INFO
8745         WRITE PRINT-LINE
8746     *
8747         MOVE SPACES TO PRINT-LINE
8748         STRING WS-VT-COVTEXT,
8749             " Motorists coverage - This coverage is afforded
8750     -     " for vehicle(s)"
8751         DELIMITED BY SIZE
8752         INTO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)
8753         ADD 1 TO WS-HIREDAUTO-COUNT
8754     *
8755         MOVE SPACES TO PRINT-LINE
8756         STRING "garaged in Vermont (only) when Hired Auto
8757     -     " coverage and ",
8758         WS-VT-COVTEXT
8759         DELIMITED BY SIZE
8760         INTO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)
8761         ADD 1 TO WS-HIREDAUTO-COUNT
8762     *
8763         MOVE SPACES TO PRINT-LINE
8764         MOVE "Motorists coverage are provided."
8765         TO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)
8766         ADD 1 TO WS-HIREDAUTO-COUNT
8767     *
8768         MOVE SPACES TO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)
8769         ADD 1 TO WS-HIREDAUTO-COUNT
8770     *
8771         PERFORM 73001-PRINT-UIUM-HIRED VARYING WS-HIREDAUTO-COUNT
8772             FROM 1 BY 1 UNTIL WS-HIREDAUTO-COUNT = 6
8773
8774
8775
8776     *****--      END-IF
8777     *****--Louisiana Only
8778     else
8779
8780         IF WS-LA-LOC = 'Y'
8781             MOVE SPACES   TO PRINT-LINE
8782             MOVE WS-12-CHAR TO PRT-MISC-INFO
8783             WRITE PRINT-LINE
8784         *
8785             MOVE SPACES TO PRINT-LINE
8786             MOVE "Uninsured Motorists Coverage - This coverage is
8787     -     " afforded for vehicle(s) garaged"
8788             TO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)

```

```

8789             ADD 1 TO WS-HIREDAUTO-COUNT
8790 *
8791             MOVE SPACES TO PRINT-LINE
8792             MOVE SPACES TO PRT-IL-COVERAGE-DESC
8793             MOVE "in Louisiana (only) when Hired Auto coverage and
8794 -             " Uninsured Motorists"
8795                 TO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)
8796                 ADD 1 TO WS-HIREDAUTO-COUNT
8797 *
8798             MOVE SPACES TO PRINT-LINE
8799             MOVE "coverage are provided."
8800                 TO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)
8801                 ADD 1 TO WS-HIREDAUTO-COUNT
8802 *
8803             MOVE SPACES TO WS-HIRED-AUTO-TEXT(WS-HIREDAUTO-COUNT)
8804             ADD 1 TO WS-HIREDAUTO-COUNT
8805 *
8806             PERFORM 73001-PRINT-UIUM-HIRED VARYING WS-HIREDAUTO-COUNT
8807                 FROM 1 BY 1 UNTIL WS-HIREDAUTO-COUNT = 4
8808
8809
8810
8811             END-IF
8812
8813 *****--
8814 *
8815             PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8816                 VARYING WS-BLANK-LINE FROM 1 BY 1
8817                 UNTIL WS-BLANK-LINE > 2
8818
8819 *
8820 *****-- HIRED AUTO INDICATOR
8821             END-IF.
8822 *****
8823 *****-- THIS IS THE NONOWNED SECTION
8824 *****
8825
8826 *****-- IF ALL THE SPECIFIED STATE VARIABLES ARE 'NO', THEN
8827 *****-- THE LINE SPACING NEEDS TO BE LIKE THE ORIGINAL VERSION.
8828 *****-- OTHERWISE, WE NEED TO DO ALL THIS CUSTOM TEXT.
8829
8830             IF WS-WI-IL-LOC EQUAL SPACES
8831             MOVE 'N' TO WS-WI-IL-LOC
8832             END-IF
8833
8834             IF WS-VT-IL-LOC EQUAL SPACES
8835             MOVE 'N' TO WS-VT-IL-LOC
8836             END-IF
8837
8838
8839             IF WS-IL-LOC  = 'N' AND
8840             WS-VT-LOC    = 'N' AND
8841             WS-WI-LOC     = 'N' AND
8842             WS-LA-LOC     = 'N' AND
8843             WS-WI-IL-LOC  = 'N' AND
8844             WS-VT-IL-LOC  = 'N'

```

```

8845
8846     PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8847     VARYING WS-BLANK-LINE FROM 1 BY 1
8848     UNTIL WS-BLANK-LINE > 13
8849
8850
8851     MOVE SPACES TO PRINT-LINE
8852     MOVE WS-SUM-WRIT-PREM-NON-OWNED
8853         TO PRT-HN-COST-DECIMAL
8854     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
8855
8856 ELSE
8857
8858     IF WS-WI-IL-LOC = 'N' AND WS-VT-IL-LOC = 'N'
8859         IF WS-NOMED-IND = 'Y'
8860             PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8861             VARYING WS-BLANK-LINE FROM 1 BY 1
8862             UNTIL WS-BLANK-LINE > 4
8863 *****-- THIS SPACES THE PREMIUM AMOUNT
8864         ELSE
8865             PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8866             VARYING WS-BLANK-LINE FROM 1 BY 1
8867             UNTIL WS-BLANK-LINE > 8
8868         END-IF
8869     ELSE
8870         PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
8871         VARYING WS-BLANK-LINE FROM 1 BY 1
8872         UNTIL WS-BLANK-LINE > 4
8873     END-IF
8874
8875     MOVE SPACES    TO PRINT-LINE
8876     MOVE WS-COURIER TO PRT-MISC-INFO
8877     WRITE PRINT-LINE AFTER ADVANCING 1 LINE
8878 *
8879     MOVE SPACES TO PRINT-LINE
8880     MOVE WS-SUM-WRIT-PREM-NON-OWNED
8881         TO PRT-HN-COST-DECIMAL
8882     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
8883 *
8884     MOVE SPACES  TO PRINT-LINE
8885     MOVE WS-12-CHAR TO PRT-MISC-INFO
8886     WRITE PRINT-LINE
8887
8888 *
8889     IF WS-NO-IND = 'Y'
8890     IF WS-VT-IL-LOC = "Y" OR WS-IL-LOC = "Y" THEN
8891         IF WS-NONOWNED-COUNT = 0
8892             MOVE SPACES    TO PRINT-LINE
8893             MOVE WS-12-CHAR TO PRT-MISC-INFO
8894             WRITE PRINT-LINE
8895         END-IF
8896 *
8897     MOVE SPACES TO PRINT-LINE
8898     STRING WS-IL-COVGTEXT,
8899         " Motorists coverage - This coverage is afforded
8900     -    " for vehicle(s)"

```

```

8901             DELIMITED BY SIZE
8902 INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8903             ADD 1 TO WS-NONOWNED-COUNT
8904 *
8905 IF WS-VT-IL-LOC = 'Y'
8906     MOVE SPACES TO PRINT-LINE
8907     STRING "garaged in ",
8908     WS-STATE-VAR22,
8909     " when Non-Owned Auto coverage and ",
8910     WS-IL-COVGTEXT
8911         DELIMITED BY SIZE
8912     INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8913     ADD 1 TO WS-NONOWNED-COUNT
8914 *
8915 ELSE
8916     IF WS-IL-LOC = 'Y'
8917         MOVE SPACES TO PRINT-LINE
8918         STRING "garaged in ",
8919         WS-STATE-VAR9,
8920         "when Non-Owned Auto coverage and ",
8921         WS-IL-COVGTEXT
8922             DELIMITED BY SIZE
8923         INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8924         ADD 1 TO WS-NONOWNED-COUNT
8925     END-IF
8926 END-IF
8927 *
8928 MOVE SPACES TO PRINT-LINE
8929 MOVE "Motorists coverage are provided."
8930 TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8931     ADD 1 TO WS-NONOWNED-COUNT
8932 *
8933     MOVE SPACES TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8934     ADD 1 TO WS-NONOWNED-COUNT
8935
8936
8937 if WS-IL-LOC = "Y"
8938 PERFORM 73002-PRINT-UIUM VARYING WS-NONOWNED-COUNT
8939     FROM 1 BY 1 UNTIL WS-NONOWNED-COUNT = 8
8940     MOVE 1 TO WS-NONOWNED-COUNT
8941 end-if
8942
8943
8944
8945 *
8946 *****-- WS-VT-IL-LOC = "Y" OR WS-IL-LOC = "Y"
8947     END-IF
8948 *****
8949
8950 IF WS-WI-LOC = "Y" THEN
8951     MOVE SPACES TO PRINT-LINE
8952     MOVE "Uninsured Motorists Coverage - This coverage is
8953 -     " afforded for vehicle(s) garaged"
8954     TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8955     ADD 1 TO WS-NONOWNED-COUNT
8956 *

```

```

8957      MOVE SPACES TO PRINT-LINE
8958      MOVE "in Wisconsin (only) when Non-Owned Auto coverage
8959  -      " and Uninsured Motorists"
8960      TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8961      ADD 1 TO WS-NONOWNED-COUNT
8962  *
8963      MOVE SPACES TO PRINT-LINE
8964      MOVE "coverage are provided."
8965      TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8966      ADD 1 TO WS-NONOWNED-COUNT
8967
8968      MOVE SPACES TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8969      ADD 1 TO WS-NONOWNED-COUNT
8970  *****
8971      IF WS-NOMED-IND = 'Y'
8972      ADD 1 TO WS-NONOWNED-COUNT
8973
8974      STRING "Medical Payments Coverage - This coverage
8975  -      " is afforded for vehicle(s) garaged in "
8976      DELIMITED BY SIZE
8977      INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8978      ADD 1 TO WS-NONOWNED-COUNT
8979  *
8980      STRING "Wisconsin (only) when Non-Owned Auto coverage
8981  -      " and Medical Payments coverage are"
8982      DELIMITED BY SIZE
8983      INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8984      ADD 1 TO WS-NONOWNED-COUNT
8985  *
8986      STRING "provided."
8987      DELIMITED BY SIZE
8988      INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8989      ADD 1 TO WS-NONOWNED-COUNT
8990  *
8991      MOVE SPACES TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
8992      ADD 1 TO WS-NONOWNED-COUNT
8993      END-IF
8994  *****
8995
8996      if ws-nomed-ind = 'Y'
8997      PERFORM 73002-PRINT-UIUM VARYING WS-NONOWNED-COUNT
8998      FROM 1 BY 1 UNTIL WS-NONOWNED-COUNT = 10
8999
9000      MOVE 'AUCOV3'    TO WS-PLFD-GB-FORM-DESC
9001      MOVE 'Y'         TO WS-PLFD-GB-ENABLE
9002      MOVE 'XX'        TO WS-PLFD-GB-OPTION-1
9003      MOVE 'XX'        TO WS-PLFD-GB-OPTION-2
9004  *
9005      MOVE 'XX' TO WS-PLFD-GB-STATE
9006      PERFORM 8500-GET-MACRO THRU 8500-EXIT
9007      else
9008      PERFORM 73002-PRINT-UIUM VARYING WS-NONOWNED-COUNT
9009      FROM 1 BY 1 UNTIL WS-NONOWNED-COUNT = 10
9010      end-if
9011  *****
9012      END-IF

```



```

9013 *****
9014 *
9015 IF WS-VT-LOC = "Y" THEN
9016     MOVE SPACES TO PRINT-LINE
9017     MOVE WS-12-CHAR TO PRT-MISC-INFO
9018 *
9019     MOVE SPACES TO PRINT-LINE
9020     STRING WS-VT-COVTXT,
9021     " Motorists coverage - This coverage is afforded
9022 -     " for vehicle(s)"
9023         DELIMITED BY SIZE
9024         INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9025         ADD 1 TO WS-NONOWNED-COUNT
9026 *
9027     MOVE SPACES TO PRINT-LINE
9028     STRING "garaged in Vermont (only) when Non-Owned Auto
9029 -     " coverage and ",
9030     WS-VT-COVTXT
9031         DELIMITED BY SIZE
9032         INTO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9033         ADD 1 TO WS-NONOWNED-COUNT
9034 *
9035     MOVE SPACES TO PRINT-LINE
9036     MOVE "Motorist coverage are provided."
9037         TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9038         ADD 1 TO WS-NONOWNED-COUNT
9039 *
9040     MOVE SPACES TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9041     ADD 1 TO WS-NONOWNED-COUNT
9042 *
9043     PERFORM 73002-PRINT-UIUM VARYING WS-NONOWNED-COUNT
9044     FROM 1 BY 1 UNTIL WS-NONOWNED-COUNT = 8
9045
9046 else
9047 IF WS-LA-LOC = 'Y' THEN
9048     MOVE SPACES TO PRINT-LINE
9049     MOVE "Uninsured Motorists Coverage - This coverage is
9050 -     " afforded for vehicle(s) garaged"
9051     TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9052     ADD 1 TO WS-NONOWNED-COUNT
9053 *
9054     MOVE SPACES TO PRINT-LINE
9055     MOVE "in Louisiana (only) when Non-Owned Auto coverage
9056 -     " and Uninsured Motorists"
9057     TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9058     ADD 1 TO WS-NONOWNED-COUNT
9059 *
9060     MOVE SPACES TO PRINT-LINE
9061     MOVE "coverage are provided."
9062     TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9063     ADD 1 TO WS-NONOWNED-COUNT
9064 *
9065     MOVE SPACES TO WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9066     ADD 1 TO WS-NONOWNED-COUNT
9067 *
9068     PERFORM 73002-PRINT-UIUM VARYING WS-NONOWNED-COUNT

```

FROM 1 BY 1 UNTIL WS-NONOWNED-COUNT = 8

*

END-IF

*****-- NON-OWNED INDICATOR

END-IF

*****-- BIG IF/AND

END-IF.

*

IF WS-WI-LOC = 'Y' AND WS-NOMED-IND = 'Y'

*****-- THIS WORKS WHEN IT IS WI AND THERE ARE MED PAYMENTS

PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT

VARYING WS-BLANK-LINE FROM 1 BY 1

UNTIL WS-BLANK-LINE > 2

ELSE

*****-- THIS IS WHEN IT IS IL/VT OR IL/WI

IF WS-WI-IL-LOC = 'Y' OR

WS-VT-IL-LOC = 'Y'

*****-- THIS WORKS WHEN THERE ARE TWO PARAGRAPHS

PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT

VARYING WS-BLANK-LINE FROM 1 BY 1

UNTIL WS-BLANK-LINE > 2

ELSE

IF WS-IL-LOC = 'N' AND

WS-VT-LOC = 'N' AND

WS-WI-LOC = 'N' AND

WS-LA-LOC = 'N'

*****-- THIS IS WHEN IT IS A STATE OTHER THAN IL/VT/WI

PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT

VARYING WS-BLANK-LINE FROM 1 BY 1

UNTIL WS-BLANK-LINE > 12

ELSE

*****-- THIS WORKS WHEN THERE IS ONLY ONE PARAGRAPH

PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT

VARYING WS-BLANK-LINE FROM 1 BY 1

UNTIL WS-BLANK-LINE > 4

END-IF

END-IF

END-IF.

```

9125         MOVE 1 TO WS-ENTRY-COUNT.
9126         MOVE 1 TO WS-LINE-COUNT.
9127     *
9128         MOVE 'N' TO WS-EOC-ENDORSE.
9129         PERFORM 7350-PRINT-ENDORSEMENT THRU 7350-EXIT
9130         UNTIL WS-EOC-ENDORSE = 'Y'.
9131     *
9132     7300-EXIT.
9133     EXIT.
9134
9135     *=====
9136     73001-PRINT-UIUM-HIRED.
9137
9138
9139         IF WS-PAGE-3A = 'Y' AND WS-HIRED-AUTO-COUNT = 9
9140         MOVE SPACES TO PRINT-LINE
9141         MOVE WS-12-CHAR TO PRT-MISC-INFO
9142         WRITE PRINT-LINE
9143         END-IF.
9144     *
9145         MOVE SPACES TO PRINT-LINE.
9146         MOVE WS-HIRED-AUTO-TEXT(WS-HIRED-AUTO-COUNT)
9147             TO PRT-IL-COVERAGE-DESC.
9148         WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
9149
9150
9151
9152
9153
9154     73001-EXIT.
9155     EXIT.
9156     *
9157     73002-PRINT-UIUM.
9158         MOVE SPACES TO PRINT-LINE.
9159         MOVE WS-NONOWNED-AUTO-TEXT(WS-NONOWNED-COUNT)
9160             TO PRT-IL-COVERAGE-DESC.
9161         WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
9162     73002-EXIT.
9163     EXIT.
9164     *
9165     *=====
9166     *
9167     7302-DETERMINE-COVG.
9168
9169         EXEC SQL AT ORA1
9170             OPEN UMUI_CURS
9171         END-EXEC.
9172
9173
9174         EXEC SQL AT ORA1
9175             OPEN DETERMINE_UMUI
9176         END-EXEC.
9177
9178
9179         EXEC SQL AT ORA1
9180             OPEN DETERMINE_TXD

```

```

9181      END-EXEC.
9182
9183
9184
9185      INITIALIZE WS-COVG-COUNT,
9186                WS-COVG-STATE,
9187                WS-AUCV-CODE,
9188                WS-AUCV-TYPE,
9189                WS-COVG-TEXT,
9190                WS-MED-IND.
9191
9192
9193
9194      EXEC SQL
9195      FETCH UMUI_CURS
9196      INTO :WS-COVG-STATE,
9197           :WS-COVG-COUNT
9198      END-EXEC.
9199
9200
9201      *
9202
9203      EXEC SQL AT ORA1
9204      SELECT 'Y' AS ACODE
9205      INTO :WS-MED-IND
9206      FROM AUTO_COVERAGE AUCV,
9207           EXPOSURE EXIT ,
9208           COVERED_ITEM COIT ,
9209           LOC LOCT
9210      WHERE AUCV_AU_NO   = EXIT_AUCV_AU_NO
9211            AND COIT_EXIT_GB_NO = EXIT_GB_NO
9212            AND COIT_LOCT_GB_NO = LOCT_GB_NO
9213            AND AUCV_AU_TYPE in ('MED','HAMED','NOMED')
9214            AND EXIT_GB_NO   IN
9215              (SELECT COIT_EXIT_GB_NO
9216               FROM COIT
9217               WHERE COIT_INFC_GB_NO = :WS-PPRT-INFC-GB-NO
9218              )
9219            AND LOCT.LOCT_STAT_GB_ALPHA_STATE_IN = :WS-LOC-STATE
9220            AND 'WI' = :WS-LOC-STATE
9221            AND ROWNUM = 1
9222      END-EXEC.
9223
9224      *
9225
9226      IF WS-MED-IND NOT EQUAL SPACES
9227          MOVE WS-MED-IND TO WS-MED-HOLD
9228      END-IF.
9229
9230      *
9231
9232      IF WS-COVG-STATE EQUAL SPACES
9233          MOVE WS-LOC-STATE TO WS-COVG-STATE
9234      END-IF.
9235
9236      IF WS-COVG-COUNT = 0
9237          MOVE 'Uninsured/Underinsured' TO WS-COVG-TEXT
9238          IF WS-COVG-STATE = 'IL'
9239              MOVE WS-COVG-TEXT TO WS-IL-COVGTEXT
9240          ELSE IF WS-COVG-STATE = 'VT'

```

```

9237         MOVE WS-COVG-TEXT TO WS-VT-COVGTEXT
9238     ELSE IF WS-COVG-STATE = 'LA'
9239         MOVE WS-COVG-TEXT TO WS-VT-COVGTEXT
9240     ELSE IF WS-COVG-STATE = 'WI'
9241         MOVE 'Uninsured' TO WS-COVG-TEXT
9242         MOVE WS-COVG-TEXT TO WS-WI-COVGTEXT
9243     end-if
9244
9245     ELSE IF WS-COVG-COUNT = 1
9246         PERFORM 7303-DETERMINE-UMUI
9247         UNTIL END-OF-UMUI-LOC
9248     ELSE IF WS-COVG-COUNT >= 2
9249         PERFORM 7303-DETERMINE-UMUI
9250         UNTIL END-OF-UMUI-LOC
9251         MOVE 'Uninsured/Underinsured' TO WS-COVG-TEXT
9252         IF WS-COVG-STATE = 'IL'
9253             MOVE WS-COVG-TEXT TO WS-IL-COVGTEXT
9254         ELSE IF WS-COVG-STATE = 'VT'
9255             MOVE WS-COVG-TEXT TO WS-VT-COVGTEXT
9256         ELSE IF WS-COVG-STATE = 'WI'
9257             MOVE WS-COVG-TEXT TO WS-WI-COVGTEXT
9258         ELSE IF WS-COVG-STATE = 'LA'
9259             MOVE WS-COVG-TEXT TO WS-LA-COVGTEXT
9260         END-IF
9261     END-IF.
9262
9263
9264 *
9265     IF WS-COVG-COUNT EQUAL 0
9266         MOVE 1 TO WS-COVG-COUNT
9267         MOVE 'Y' TO WS-HA-IND
9268         MOVE 'Y' TO WS-NO-IND
9269     END-IF.
9270 *
9271     IF WS-COVG-COUNT NOT EQUAL ZERO
9272         IF WS-COVG-STATE = 'IL'
9273             MOVE 'Y' TO WS-IL-LOC
9274         ELSE IF WS-COVG-STATE = 'WI'
9275             MOVE 'Y' TO WS-WI-LOC
9276         ELSE IF WS-COVG-STATE = 'VT'
9277             MOVE 'Y' TO WS-VT-LOC
9278         END-IF
9279     END-IF.
9280
9281 7302-EXIT.
9282  EXIT.
9283 *****
9284
9285 7303-DETERMINE-UMUI.
9286
9287     EXEC SQL
9288     FETCH DETERMINE_UMUI
9289     INTO :WS-AUCV-CODE,
9290         :WS-AUCV-TYPE
9291     END-EXEC
9292

```

```

9293         IF SQLCODE = 1403 THEN
9294             SET END-OF-UMUI-LOC TO TRUE
9295         END-IF.
9296
9297
9298         IF WS-AUCV-TYPE EQUAL SPACES
9299             MOVE WS-AUCV-CODE TO WS-AUCV-CODE
9300         END-IF.
9301
9302
9303
9304         IF WS-AUCV-CODE = 'HA'
9305             MOVE 'Y' TO WS-HA-IND
9306         ELSE IF
9307             WS-AUCV-CODE = 'NO'
9308             MOVE 'Y' TO WS-NO-IND
9309         END-IF.
9310
9311         IF WS-AUCV-TYPE = 'UM'
9312             MOVE 'Uninsured' TO WS-COVG-TEXT
9313         ELSE
9314             IF WS-AUCV-TYPE = 'UI'
9315                 MOVE 'Uninsured/Underinsured' TO WS-COVG-TEXT
9316             END-IF.
9317 *
9318         IF WS-COVG-STATE = 'IL'
9319             MOVE WS-COVG-TEXT TO WS-IL-COVGTEXT
9320         ELSE IF WS-COVG-STATE = 'VT'
9321             MOVE WS-COVG-TEXT TO WS-VT-COVGTEXT
9322         ELSE IF WS-COVG-STATE = 'WI'
9323             MOVE WS-COVG-TEXT TO WS-WI-COVGTEXT
9324         ELSE IF WS-COVG-STATE = 'LA'
9325             MOVE WS-COVG-TEXT TO WS-LA-COVGTEXT
9326         END-IF.
9327
9328
9329 7303-EXIT.
9330     EXIT.
9331
9332 *****
9333 7305-DETERMINE-TXSD.
9334
9335     EXEC SQL
9336     FETCH DETERMINE_TXD
9337     INTO :WS-TXSD-CLASS,
9338         :WS-TXDS-PX,
9339         :WS-TXDS-LIMIT
9340     END-EXEC.
9341
9342     IF SQLCODE NOT EQUAL 0
9343         AND SQLCODE NOT EQUAL 1403
9344         MOVE "ERROR SELECTING TXDS 7305 " TO WS-ERR-MSG
9345         PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
9346     else
9347         IF SQLCODE = 1403 THEN
9348             SET END-OF-TXSD TO TRUE

```



```

9349 ELSE
9350 IF WS-TXSD-CLASS = 'TXSDM' THEN
9351 IF WS-PG2-LINECNT >= 38 THEN
9352 PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
9353 END-IF
9354 MOVE SPACES TO PRINT-LINE
9355 MOVE "TEXAS SUPPLEMENTARY DEATH BENEFIT "
9356 TO PRT-COVERAGE-DESC
9357 MOVE " PER PERSON" TO PRT-COVERAGE-TEXT
9358 MOVE WS-TXDS-LIMIT TO PRT-COVERAGE-LIMIT
9359 MOVE WS-TXDS-PX TO PRT-ZERO-COST-DECIMAL
9360 WRITE PRINT-LINE AFTER ADVANCING 2 LINES
9361 ADD 2 TO WS-PG2-LINECNT
9362
9363 MOVE SPACES TO PRINT-LINE
9364 MOVE " MEDICAL PAYMENTS " TO PRT-COVERAGE-DESC
9365 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9366 ADD 1 TO WS-PG2-LINECNT
9367 END-IF
9368 *
9369 IF WS-TXSD-CLASS = 'TXSDP' THEN
9370 IF WS-PG2-LINECNT >= 38 THEN
9371 PERFORM 7275-PRINT-PAGE-2A THRU 7275-EXIT
9372 END-IF
9373 MOVE SPACES TO PRINT-LINE
9374 MOVE "TEXAS SUPPLEMENTARY DEATH BENEFIT "
9375 TO PRT-COVERAGE-DESC
9376 MOVE " PER PERSON" TO PRT-COVERAGE-TEXT
9377 MOVE WS-TXDS-LIMIT TO PRT-COVERAGE-LIMIT
9378 MOVE WS-TXDS-PX TO PRT-ZERO-COST-DECIMAL
9379 WRITE PRINT-LINE AFTER ADVANCING 2 LINES
9380 ADD 2 TO WS-PG2-LINECNT
9381
9382 MOVE SPACES TO PRINT-LINE
9383 MOVE " PERSONAL INJURY PROTECTION "
9384 TO PRT-COVERAGE-DESC
9385 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9386 ADD 1 TO WS-PG2-LINECNT
9387 END-IF
9388
9389 END-IF
9390 END-IF.
9391 *
9392 7305-EXIT.
9393 EXIT.
9394 *****
9395 *=====*
9396
9397 *-----*
9398 * THIS PARAGRAPH WILL PRINT THE ENDORSEMENTS ON PAGE 3 ITEM SIX *
9399 * OF THE AUTO POLICY. *
9400 *-----*
9401 *
9402 7350-PRINT-ENDORSEMENT.
9403
9404 IF WS-LINE-COUNT > 5

```

```

9405      MOVE SPACES      TO WS-PRINT-MACRO-LINE
9406      MOVE ESC          TO WS-PRINT-ESC
9407      MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO
9408      MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE
9409      WRITE PRINT-LINE AFTER ADVANCING PAGE
9410      MOVE SPACES      TO PRINT-LINE
9411      MOVE WS-LINEPTR   TO PRT-MISC-INFO
9412      WRITE PRINT-LINE
9413      MOVE 1 TO WS-LINE-COUNT
9414  ELSE
9415      if ws-med-hold NOT EQUAL 'Y'
9416          MOVE SPACES      TO PRINT-LINE
9417          MOVE WS-LINEPTR   TO PRT-MISC-INFO
9418          WRITE PRINT-LINE
9419      else if ws-med-hold = 'Y' and WS-PAGE-3A NOT EQUAL 'Y'
9420          MOVE SPACES      TO PRINT-LINE
9421          MOVE WS-LINEPTR   TO PRT-MISC-INFO
9422          WRITE PRINT-LINE after advancing 1 line
9423      end-if
9424  end-if
9425  END-IF.
9426  *
9427      MOVE WS-PLFD-GB-FORM-NO TO WS-HOLD-FORM-NO.
9428      MOVE 0 TO WS-LINE-COUNT.
9429  *
9430      PERFORM 7375-PRINT-ENDORSEMENT-LINE THRU 7375-EXIT
9431      UNTIL WS-EOC-ENDORSE = 'Y' .
9432  *
9433      IF WS-PAGE-3A = 'Y' AND WS-LINE-COUNT < 53 THEN
9434          PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9435          UNTIL WS-LINE-COUNT = 53
9436      ELSE
9437          IF WS-LINE-COUNT < 6 THEN
9438              PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9439              UNTIL WS-LINE-COUNT = 6
9440          END-IF
9441      END-IF.
9442  *
9443      MOVE SPACES      TO PRINT-LINE.
9444      MOVE WS-COURIER   TO PRT-MISC-INFO.
9445      WRITE PRINT-LINE  AFTER ADVANCING 1 LINE.
9446
9447  *****
9448      IF WS-PAGE-3A NOT EQUAL 'Y'
9449          IF WS-NOMED-IND = 'Y'
9450              MOVE SPACES TO PRINT-LINE
9451              MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-5-INFO
9452              WRITE PRINT-LINE
9453                  AFTER ADVANCING 2 LINES
9454          ELSE
9455              MOVE SPACES TO PRINT-LINE
9456              MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-5-INFO
9457              WRITE PRINT-LINE
9458                  AFTER ADVANCING 2 LINES
9459          END-IF
9460      END-IF.

```

```

9461
9462 *****
9463 IF WS-EOC-ENDORSE = 'N'
9464 IF WS-PAGE-3A = 'Y'
9465     MOVE 'AUCOV3' TO WS-PLFD-GB-FORM-DESC
9466     MOVE 'Y' TO WS-PLFD-GB-ENABLE
9467     MOVE 'XX' TO WS-PLFD-GB-OPTION-1
9468     MOVE 'XX' TO WS-PLFD-GB-OPTION-2
9469 *
9470     MOVE 'XX' TO WS-PLFD-GB-STATE
9471     PERFORM 8500-GET-MACRO THRU 8500-EXIT
9472 *
9473     IF SQLCODE NOT EQUAL 0
9474         MOVE "ERROR SELECTING MACRO FOR STATE XX - 7350"
9475         TO WS-ERR-MSG
9476         PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
9477     END-IF
9478 *
9479     MOVE SPACES TO WS-PRINT-MACRO-LINE
9480     MOVE ESC TO WS-PRINT-ESC
9481     MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO
9482     MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE
9483     WRITE PRINT-LINE AFTER ADVANCING PAGE
9484 *
9485     MOVE WS-PLFD-GB-FORM-NO TO WS-HOLD-FORM-NO
9486     MOVE 0 TO WS-LINE-COUNT
9487 *
9488     MOVE SPACES TO PRINT-LINE
9489     MOVE WS-LINEPTR TO PRT-MISC-INFO
9490     WRITE PRINT-LINE
9491 *
9492     PERFORM 7375-PRINT-ENDORSEMENT-LINE THRU 7375-EXIT
9493     UNTIL WS-EOC-ENDORSE = 'Y'
9494 END-IF
9495 END-IF.
9496
9497 *
9498 7350-EXIT.
9499 EXIT.
9500 *
9501 7375-PRINT-ENDORSEMENT-LINE.
9502
9503 *
9504 IF WS-END-1(WS-ENTRY-COUNT) EQUAL SPACES
9505     MOVE 'Y' TO WS-EOC-ENDORSE
9506 END-IF.
9507 *
9508
9509 IF WS-PAGE-3A = 'Y' AND WS-HIREDAUTO-COUNT = 8
9510     if ws-hamed-ind = 'Y'
9511     PERFORM 73001-PRINT-UIUM-HIRED
9512         VARYING WS-HIREDAUTO-COUNT
9513         FROM 9 BY 1 UNTIL WS-HIREDAUTO-COUNT = 12
9514
9515 *
9516     PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT

```

```

9517             VARYING WS-BLANK-LINE FROM 1 BY 1
9518             UNTIL WS-BLANK-LINE > 19
9519         end-if
9520     END-IF.
9521 *
9522     IF WS-PAGE-3A = 'Y' AND WS-NONOWNED-COUNT = 10
9523         if ws-nomed-ind = 'Y'
9524             PERFORM 73002-PRINT-UIUM
9525                 VARYING WS-NONOWNED-COUNT
9526                 FROM 9 BY 1 UNTIL WS-NONOWNED-COUNT = 13
9527         end-if
9528     END-IF.
9529 *
9530     IF WS-PAGE-3A = 'Y' AND WS-ENTRY-COUNT = 7
9531         if WS-MED-HOLD = 'Y'
9532             PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9533                 VARYING WS-BLANK-LINE FROM 1 BY 1
9534                 UNTIL WS-BLANK-LINE > 19
9535             MOVE SPACES      TO PRINT-LINE
9536             MOVE WS-LINEPTR   TO PRT-MISC-INFO
9537             WRITE PRINT-LINE
9538         else
9539             PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9540                 VARYING WS-BLANK-LINE FROM 1 BY 1
9541                 UNTIL WS-BLANK-LINE > 42
9542             MOVE SPACES      TO PRINT-LINE
9543             MOVE WS-LINEPTR   TO PRT-MISC-INFO
9544             WRITE PRINT-LINE
9545         end-if
9546
9547
9548
9549
9550
9551     END-IF.
9552 *
9553     IF WS-EOC-ENDORSE = 'N' THEN
9554         MOVE SPACES TO PRINT-LINE
9555         MOVE WS-END-1(WS-ENTRY-COUNT) TO PRT-END-1
9556         MOVE WS-END-2(WS-ENTRY-COUNT) TO PRT-END-2
9557         MOVE WS-END-3(WS-ENTRY-COUNT) TO PRT-END-3
9558         MOVE WS-END-4(WS-ENTRY-COUNT) TO PRT-END-4
9559         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9560 *
9561         ADD 1 TO WS-ENTRY-COUNT
9562         ADD 1 TO WS-LINE-COUNT
9563 *
9564     IF (WS-LINE-COUNT = 5 AND WS-PAGE-3A = 'N'
9565         AND WS-END-1(WS-ENTRY-COUNT) NOT EQUAL SPACES) OR
9566        (WS-LINE-COUNT = 53 AND WS-PAGE-3A = 'Y') THEN
9567 *
9568         MOVE 'Y'      TO WS-PAGE-3A
9569         MOVE 'AUCOV3' TO WS-PLFD-GB-FORM-DESC
9570         MOVE 'Y'      TO WS-PLFD-GB-ENABLE
9571         MOVE 'XX'     TO WS-PLFD-GB-OPTION-1
9572         MOVE 'XX'     TO WS-PLFD-GB-OPTION-2

```

```

9573      MOVE 'XX' TO WS-PLFD-GB-STATE
9574      PERFORM 8500-GET-MACRO THRU 8500-EXIT
9575      *
9576      IF SQLCODE NOT EQUAL 0
9577          MOVE "ERROR SELECTING MACRO FOR STATE XX - 7375"
9578          TO WS-ERR-MSG
9579          PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
9580      END-IF
9581
9582
9583      IF WS-EOC-ENDORSE = 'N' THEN
9584          MOVE SPACES TO PRINT-LINE
9585          MOVE WS-END-1(WS-ENTRY-COUNT) TO PRT-END-1
9586          MOVE WS-END-2(WS-ENTRY-COUNT) TO PRT-END-2
9587          MOVE WS-END-3(WS-ENTRY-COUNT) TO PRT-END-3
9588          MOVE WS-END-4(WS-ENTRY-COUNT) TO PRT-END-4
9589          WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9590      *
9591          ADD 1 TO WS-ENTRY-COUNT
9592          ADD 1 TO WS-LINE-COUNT
9593      *
9594          MOVE SPACES TO PRINT-LINE
9595          MOVE WS-COURIER TO PRT-MISC-INFO
9596          WRITE PRINT-LINE AFTER ADVANCING 1 LINE
9597      *
9598          MOVE SPACES TO PRINT-LINE
9599          MOVE WS-HOLD-FORM-NO TO PRT-MISC-5-INFO
9600
9601          WRITE PRINT-LINE
9602          AFTER ADVANCING 2 LINES
9603      *
9604          MOVE SPACES TO WS-PRINT-MACRO-LINE
9605          MOVE ESC TO WS-PRINT-ESC
9606          MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO
9607          MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE
9608          WRITE PRINT-LINE AFTER ADVANCING PAGE
9609      *
9610          if ws-med-hold not equal 'Y'
9611              MOVE SPACES TO PRINT-LINE
9612              PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9613              VARYING WS-BLANK-LINE FROM 1 BY 1
9614              UNTIL WS-BLANK-LINE > 5
9615              MOVE 1 TO WS-LINE-COUNT
9616          else
9617              MOVE SPACES TO PRINT-LINE
9618              PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9619              VARYING WS-BLANK-LINE FROM 1 BY 1
9620              UNTIL WS-BLANK-LINE > 1
9621              MOVE 1 TO WS-LINE-COUNT
9622      *          end-if
9623      END-IF
9624      END-IF.
9625
9626
9627      *
9628      7375-EXIT.

```

```

9629      EXIT.
9630      *
9631      *-----*
9632      *  THIS PARAGRAPH WILL PRINT THE SCHEDULE OF AUTOMOBILES AND      *
9633      *  SCHEDULE OF PREM ON PAGE 5 OF THE AUTO POLICY.                *
9634      *-----*
9635      *
9636      7400-PRINT-SCHEDULE-PAGE.
9637      *
9638      MOVE 'AUSCH' TO WS-PLFD-GB-FORM-DESC.
9639      MOVE 'Y'   TO WS-PLFD-GB-ENABLE.
9640      MOVE 'XX'  TO WS-PLFD-GB-OPTION-1.
9641      MOVE 'O'   TO WS-PLFD-GB-OPTION-2.
9642      MOVE WS-TBL-STATE(WS-ENTRY-COUNT) TO WS-PLFD-GB-STATE
9643      *
9644      PERFORM 8500-GET-MACRO THRU 8500-EXIT.
9645      *
9646      IF SQLCODE EQUAL 1403
9647      MOVE 'XX' TO WS-PLFD-GB-STATE
9648      PERFORM 8500-GET-MACRO THRU 8500-EXIT
9649      END-IF.
9650      *
9651      IF SQLCODE NOT EQUAL 0
9652      MOVE "ERROR SELECTING MACRO FOR STATE XX - 7400"
9653      TO WS-ERR-MSG
9654      PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
9655      END-IF.
9656      *
9657      MOVE SPACES      TO WS-PRINT-MACRO-LINE.
9658      MOVE ESC         TO WS-PRINT-ESC.
9659      MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO.
9660      MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE.
9661      WRITE PRINT-LINE AFTER ADVANCING PAGE.
9662      *
9663      MOVE SPACES      TO PRINT-LINE.
9664      MOVE WS-LINEPTR   TO PRT-MISC-INFO.
9665      WRITE PRINT-LINE.
9666      *
9667      MOVE SPACES      TO PRINT-LINE.
9668      MOVE WS-POLY-NO   TO PRT-MISC-1-INFO.
9669      IF WS-PLFD-GB-OPTION-OUT = 'O'
9670      WRITE PRINT-LINE
9671      ELSE
9672      WRITE PRINT-LINE  AFTER ADVANCING 1 LINES
9673      END-IF.
9674      *
9675      if WS-POLY-BSOR-GB-FM-ACCT = 5
9676      MOVE SPACES TO PRINT-LINE
9677      MOVE WS-CLIENT-NAME TO PRT-MISC-SN2-INFO
9678      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9679      else
9680      MOVE SPACES TO PRINT-LINE
9681      MOVE WS-CLIENT-NAME TO PRT-MISC-2-INFO
9682      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9683      end-if.
9684

```



```

9685
9686 *
9687 IF WS-FIRST-TIME = 'Y'
9688     MOVE 1 TO WS-ENTRY-COUNT-3
9689 END-IF.
9690 *
9691 PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9692 VARYING WS-BLANK-LINE FROM 1 BY 1
9693     UNTIL WS-BLANK-LINE > 5.
9694 *
9695 MOVE WS-TBL-STATE(WS-ENTRY-COUNT) TO WS-HOLD-STATE.
9696 MOVE WS-ENTRY-COUNT TO WS-HOLD-ENTRY-COUNT.
9697 *
9698 IF WS-PLFD-GB-OPTION-OUT = 'O'
9699     PERFORM 7475-WRITE-AUTO-INFO THRU 7475-EXIT
9700     VARYING WS-ENTRY-COUNT-2 FROM 1 BY 1
9701     UNTIL WS-ENTRY-COUNT-2 = 11
9702 ELSE
9703     PERFORM 7425-WRITE-AUTO-INFO THRU 7425-EXIT
9704     VARYING WS-ENTRY-COUNT-2 FROM 1 BY 1
9705     UNTIL WS-ENTRY-COUNT-2 = 11
9706 END-IF.
9707 *
9708 MOVE WS-HOLD-ENTRY-COUNT TO WS-ENTRY-COUNT.
9709 *
9710 PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
9711 VARYING WS-BLANK-LINE FROM 1 BY 1
9712     UNTIL WS-BLANK-LINE > 8.
9713 *
9714 PERFORM 7450-WRITE-PREM-INFO THRU 7450-EXIT
9715 VARYING WS-ENTRY-COUNT-2 FROM 1 BY 1
9716     UNTIL WS-ENTRY-COUNT-2 = 11.
9717 *
9718 MOVE 'N' TO WS-FIRST-TIME.
9719 *
9720 MOVE SPACES TO PRINT-LINE.
9721 MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-6-INFO.
9722 WRITE PRINT-LINE AFTER ADVANCING 3 LINES.
9723 *
9724 7400-EXIT.
9725 EXIT.
9726 *
9727 *-----*
9728 * THIS PARAGRAPH WILL PRINT THE SCHEDULE OF AUTOMOBILES. THE *
9729 * PARAGRAPH WILL BE PERFORMED 10 TIMES. THEN THE 7400 PARAGRAPH *
9730 * WILL CONTINUE. IF MORE THAN 10 AUTO EXISTS, THIS PARAGRAPH WILL *
9731 * BE PERFORMED 10 TIMES AGAIN. THIS INFORMATION WILL BE PRINTED *
9732 * ON THE NEW SCHEDULE OF AUTOMOBILES PAGES. *
9733 *-----*
9734 *
9735 7425-WRITE-AUTO-INFO.
9736 *
9737 MOVE SPACES TO PRINT-LINE.
9738 *
9739 IF WS-TBL-VHDS-AU-NO(WS-ENTRY-COUNT) EQUAL 0
9740     OR WS-TBL-STATE(WS-ENTRY-COUNT) NOT EQUAL WS-HOLD-STATE

```

```

9741         MOVE SPACES TO PRINT-LINE
9742         WRITE PRINT-LINE AFTER ADVANCING 2 LINES
9743     ELSE
9744 *
9745         STRING WS-TBL-CITY(WS-ENTRY-COUNT)(1:14)
9746             DELIMITED BY " "
9747             "/"
9748             DELIMITED BY SIZE
9749             INTO PRT-POP-DEN
9750         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9751         MOVE SPACES TO PRINT-LINE
9752 *
9753         MOVE WS-TBL-ID-NO(WS-ENTRY-COUNT) TO PRT-AUTO-NO
9754         MOVE WS-TBL-YEAR(WS-ENTRY-COUNT)(1:4) TO PRT-YEAR
9755         STRING WS-TBL-MAKE(WS-ENTRY-COUNT),
9756             " ",
9757             WS-TBL-MODEL(WS-ENTRY-COUNT)
9758             DELIMITED BY " "
9759         INTO PRT-MAKE-MODEL
9760         MOVE WS-TBL-SERIAL-NO(WS-ENTRY-COUNT) TO PRT-SERIAL-NO
9761         MOVE WS-TBL-POP-DENSITY(WS-ENTRY-COUNT) TO PRT-POP-DEN
9762         IF WS-TBL-DRIVER-REC(WS-ENTRY-COUNT)EQUAL SPACES THEN
9763             MOVE " NA " TO PRT-DRIVER-REC
9764         ELSE
9765             MOVE WS-TBL-DRIVER-REC(WS-ENTRY-COUNT)
9766             TO PRT-DRIVER-REC
9767         END-IF
9768         if WS-TBL-CLASS-CODE(WS-ENTRY-COUNT) not equal '7960'
9769         if WS-TBL-CLASS-CODE(WS-ENTRY-COUNT) not equal '7961'
9770         if WS-TBL-CLASS-CODE(WS-ENTRY-COUNT) not equal '7963'
9771
9772         IF WS-TBL-USAGE-TYPE(WS-ENTRY-COUNT) = 'C'
9773         AND WS-TBL-MILEAGE(WS-ENTRY-COUNT) EQUAL SPACES
9774             IF WS-TBL-CLASS(WS-ENTRY-COUNT) = '23'
9775
9776                 EVALUATE WS-TBL-CLASS-CODE(WS-ENTRY-COUNT)(3:1)
9777                     WHEN '1'
9778                         MOVE 'LOCAL' TO PRT-MILEAGE
9779                     WHEN '2'
9780                         MOVE 'INTER' TO PRT-MILEAGE
9781                     WHEN '3'
9782                         MOVE 'LONG' TO PRT-MILEAGE
9783                     WHEN '4'
9784                         MOVE 'LOCAL' TO PRT-MILEAGE
9785                     WHEN '5'
9786                         MOVE 'INTER' TO PRT-MILEAGE
9787                     WHEN '6'
9788                         MOVE 'LONG' TO PRT-MILEAGE
9789                 END-EVALUATE
9790             ELSE
9791                 IF WS-TBL-CLASS(WS-ENTRY-COUNT) = '26'
9792                     MOVE 'LONG' TO PRT-MILEAGE
9793                 ELSE
9794                     MOVE " NA" TO PRT-MILEAGE
9795                 END-IF
9796             END-IF

```

```

9797         ELSE
9798             MOVE WS-TBL-MILEAGE(WS-ENTRY-COUNT) TO PRT-MILEAGE
9799         END-IF
9800     end-if
9801     end-if
9802     end-if
9803         IF PRT-MILEAGE EQUAL SPACES
9804             MOVE " NA" TO PRT-MILEAGE
9805         END-IF
9806         MOVE WS-TBL-USE(WS-ENTRY-COUNT)    TO PRT-USE
9807         MOVE WS-TBL-WEIGHT(WS-ENTRY-COUNT) TO PRT-WEIGHT
9808         MOVE WS-TBL-COST(WS-ENTRY-COUNT)   TO PRT-CAR-COST
9809         MOVE WS-TBL-LOSS-PAYEE(WS-ENTRY-COUNT) TO PRT-LOSS-PAYEE
9810         ADD 1 TO WS-ENTRY-COUNT
9811         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
9812     END-IF.
9813 *
9814 7425-EXIT.
9815     EXIT.
9816 *
9817 *-----*
9818 * THIS PARAGRAPH WILL PRINT THE SCHEDULE OF PREM. THE *
9819 * PARAGRAPH WILL BE PERFORMED 10 TIMES. THEN THE 7400 PARAGRAPH *
9820 * WILL CONTINUE. IF MORE THAN 10 AUTO EXISTS, THIS PARAGRAPH WILL *
9821 * BE PERFORMED 10 TIMES AGAIN. *
9822 *-----*
9823 *
9824 7450-WRITE-PREM-INFO.
9825 *
9826     MOVE SPACES TO PRINT-LINE.
9827 *
9828     IF WS-TBL-VHDS-AU-NO(WS-ENTRY-COUNT) EQUAL 0
9829         OR WS-TBL-STATE(WS-ENTRY-COUNT) NOT EQUAL WS-HOLD-STATE
9830             MOVE SPACES TO PRINT-LINE
9831     ELSE
9832         MOVE WS-TBL-ID-NO(WS-ENTRY-COUNT) TO PRT-PREM-NO
9833         IF WS-HOLD-STATE = 'FL'
9834             AND WS-POLY-EFF-DATE-YYYYMMDD >= '20091001' THEN
9835                 PERFORM 7451-FORMAT-SCHED-FL THRU 7451-EXIT
9836             ELSE
9837                 PERFORM 7452-FORMAT-SCHED-LEFT THRU 7452-EXIT
9838                 PERFORM 7453-FORMAT-SCHED-RIGHT THRU 7453-EXIT
9839             END-IF
9840         END-IF.
9841 *
9842     WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
9843 *
9844 7450-EXIT.
9845     EXIT.
9846 *
9847 *-----*
9848 * THIS PARGRAPH WILL FILL THE PREM SCHEDULE FOR FL *
9849 *-----*
9850 *
9851 7451-FORMAT-SCHED-FL.
9852 *

```

```

9853      IF WS-LIAB-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9854          MOVE WS-TBL-LIAB-WRIT(WS-ENTRY-COUNT)
9855              TO PRT-FL-ZERO-LIAB-WRIT
9856      ELSE
9857          MOVE WS-TBL-LIAB-WRIT(WS-ENTRY-COUNT)
9858              TO PRT-FL-LIAB-WRIT
9859      END-IF
9860      IF WS-PIP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
9861          WS-OBEL-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9862          MOVE WS-TBL-PIP-WRIT(WS-ENTRY-COUNT)
9863              TO PRT-FL-ZERO-PIP-WRIT
9864      ELSE
9865          MOVE WS-TBL-PIP-WRIT(WS-ENTRY-COUNT)
9866              TO PRT-FL-PIP-WRIT
9867      END-IF
9868      IF WS-ADPIP-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9869          MOVE WS-TBL-ADDED-PIP-WRIT(WS-ENTRY-COUNT)
9870              TO PRT-FL-ZERO-ADDED-PIP-WRIT
9871      ELSE
9872          MOVE WS-TBL-ADDED-PIP-WRIT(WS-ENTRY-COUNT)
9873              TO PRT-FL-ADDED-PIP-WRIT
9874      END-IF
9875      IF WS-MED-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9876          MOVE WS-TBL-MED-PAY-WRIT(WS-ENTRY-COUNT)
9877              TO PRT-FL-ZERO-MED-PAY-WRIT
9878      ELSE
9879          MOVE WS-TBL-MED-PAY-WRIT(WS-ENTRY-COUNT)
9880              TO PRT-FL-MED-PAY-WRIT
9881      END-IF
9882      PERFORM 7453-FORMAT-SCHED-RIGHT THRU 7453-EXIT.
9883  *
9884      7451-EXIT.
9885      EXIT.
9886  *-----*
9887  *   THIS PARAGRAPH WILL FILL THE LEFT OF THE PREM SCHEDULE   *
9888  *-----*
9889  *
9890      7452-FORMAT-SCHED-LEFT.
9891  *
9892      IF WS-LIAB-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9893          MOVE WS-TBL-LIAB-WRIT(WS-ENTRY-COUNT)
9894              TO PRT-ZERO-LIAB-WRIT
9895      ELSE
9896          MOVE WS-TBL-LIAB-WRIT(WS-ENTRY-COUNT)
9897              TO PRT-LIAB-WRIT
9898      END-IF
9899      IF WS-BI-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9900          MOVE WS-TBL-BI-WRIT(WS-ENTRY-COUNT)
9901              TO PRT-ZERO-BI-WRIT
9902      ELSE
9903          MOVE WS-TBL-BI-WRIT(WS-ENTRY-COUNT) TO PRT-BI-WRIT
9904      END-IF
9905      IF WS-PD-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9906          MOVE WS-TBL-PD-WRIT(WS-ENTRY-COUNT)
9907              TO PRT-ZERO-PD-WRIT
9908      ELSE

```

```
9909         MOVE WS-TBL-PD-WRIT(WS-ENTRY-COUNT) TO PRT-PD-WRIT
9910     END-IF
9911     IF WS-HOLD-STATE = 'VA'
9912         AND WS-POLY-EFF-DATE-YYYYMMDD >= '20040201' THEN
9913         IF WS-MEDEXP-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9914             MOVE WS-TBL-MEDEXP(WS-ENTRY-COUNT)
9915                 TO PRT-ZERO-PIP-WRIT
9916         ELSE
9917             MOVE WS-TBL-MEDEXP(WS-ENTRY-COUNT) TO PRT-PIP-WRIT
9918         END-IF
9919     ELSE
9920         IF WS-NY-PIP-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9921             MOVE WS-TBL-NY-PIP-WRIT(WS-ENTRY-COUNT)
9922                 TO PRT-ZERO-PIP-WRIT
9923         ELSE
9924             IF WS-PIP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
9925                 WS-OBEL-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9926                 MOVE WS-TBL-PIP-WRIT(WS-ENTRY-COUNT)
9927                     TO PRT-ZERO-PIP-WRIT
9928             ELSE
9929                 MOVE WS-TBL-PIP-WRIT(WS-ENTRY-COUNT)
9930                     TO PRT-PIP-WRIT
9931             END-IF
9932         END-IF
9933     END-IF
9934     IF WS-ADPIP-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9935         MOVE WS-TBL-ADDED-PIP-WRIT(WS-ENTRY-COUNT)
9936             TO PRT-ZERO-ADDED-PIP-WRIT
9937     ELSE
9938         MOVE WS-TBL-ADDED-PIP-WRIT(WS-ENTRY-COUNT)
9939             TO PRT-ADDED-PIP-WRIT
9940     END-IF
9941     IF WS-HOLD-STATE = 'VA'
9942         AND WS-POLY-EFF-DATE-YYYYMMDD >= '20040201' THEN
9943         IF WS-ILB-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9944             MOVE WS-TBL-ILB(WS-ENTRY-COUNT)
9945                 TO PRT-ZERO-PPI-WRIT
9946         ELSE
9947             MOVE WS-TBL-ILB(WS-ENTRY-COUNT) TO PRT-PPI-WRIT
9948         END-IF
9949     ELSE
9950         IF WS-PPI-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9951             MOVE WS-TBL-PPI-WRIT(WS-ENTRY-COUNT)
9952                 TO PRT-ZERO-PPI-WRIT
9953         ELSE
9954             MOVE WS-TBL-PPI-WRIT(WS-ENTRY-COUNT)
9955                 TO PRT-PPI-WRIT
9956         END-IF
9957     END-IF
9958     IF WS-MED-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9959         MOVE WS-TBL-MED-PAY-WRIT(WS-ENTRY-COUNT)
9960             TO PRT-ZERO-MED-PAY-WRIT
9961     ELSE
9962         MOVE WS-TBL-MED-PAY-WRIT(WS-ENTRY-COUNT)
9963             TO PRT-MED-PAY-WRIT
9964     END-IF.
```

```

9965 7452-EXIT.
9966 EXIT.
9967 *-----*
9968 * THIS PARAGRAPH WILL FILL THE RIGHT OF THE PREM SCHEDULE LINE *
9969 *-----*
9970 *
9971 7453-FORMAT-SCHED-RIGHT.
9972 *
9973     IF WS-UM-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9974         MOVE WS-TBL-UM-WRIT(WS-ENTRY-COUNT)
9975             TO PRT-ZERO-UM-WRIT
9976     ELSE
9977         IF WS-SUM-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9978             MOVE WS-TBL-SUM-WRIT(WS-ENTRY-COUNT)
9979                 TO PRT-ZERO-UM-WRIT
9980         ELSE
9981             MOVE WS-TBL-UM-WRIT(WS-ENTRY-COUNT)
9982                 TO PRT-UM-WRIT
9983         END-IF
9984     END-IF
9985     MOVE WS-TBL-STATED-AMT(WS-ENTRY-COUNT) TO PRT-STATED-AMT
9986     MOVE WS-TBL-COMP-DEDUCT(WS-ENTRY-COUNT)
9987     TO PRT-COMP-DEDUCT
9988     IF WS-COMP-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9989         MOVE WS-TBL-COMP-WRIT(WS-ENTRY-COUNT)
9990             TO PRT-ZERO-COMP-WRIT
9991     ELSE
9992         MOVE WS-TBL-COMP-WRIT(WS-ENTRY-COUNT)
9993             TO PRT-COMP-WRIT
9994     END-IF
9995     IF WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
9996         MOVE WS-TBL-SPEC-LOSS-WRIT(WS-ENTRY-COUNT)
9997             TO PRT-ZERO-SPEC-LOSS-WRIT
9998     ELSE
9999         MOVE WS-TBL-SPEC-LOSS-WRIT(WS-ENTRY-COUNT)
10000             TO PRT-SPEC-LOSS-WRIT
10001     END-IF
10002     IF WS-LIMF(WS-ENTRY-COUNT) = 'Y' THEN
10003         MOVE WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
10004             TO PRT-ZERO-COLL-DEDUCT
10005     ELSE
10006         MOVE WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
10007             TO PRT-COLL-DEDUCT
10008     END-IF
10009     MOVE WS-TBL-WAIVER(WS-ENTRY-COUNT) TO PRT-WAIVER
10010     IF WS-COLL-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
10011         MOVE WS-TBL-COLL-WRIT(WS-ENTRY-COUNT)
10012             TO PRT-ZERO-COLL-WRIT
10013     ELSE
10014         MOVE WS-TBL-COLL-WRIT(WS-ENTRY-COUNT)
10015             TO PRT-COLL-WRIT
10016     END-IF
10017     IF WS-TOW-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
10018         MOVE WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
10019             TO PRT-ZERO-TOW-WRIT
10020     ELSE

```



```

10021         MOVE WS-TBL-TOW-WRIT(WS-ENTRY-COUNT) TO PRT-TOW-WRIT
10022     END-IF
10023     IF WS-SLL-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10024         WS-PIP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10025         WS-OBEL-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10026         WS-ADPIP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10027         WS-MED-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10028         WS-MEDEXP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10029         WS-UM-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10030         WS-UI-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10031         WS-COMP-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10032         WS-COLL-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10033         WS-SPEC-LOSS-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10034         WS-PPI-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10035         WS-ILB-EXIST(WS-ENTRY-COUNT) = 'Y' OR
10036         WS-TOW-EXIST(WS-ENTRY-COUNT) = 'Y' THEN
10037         MOVE WS-TBL-TL-PREM(WS-ENTRY-COUNT)
10038             TO PRT-ZERO-TL-PREM
10039     ELSE
10040         MOVE WS-TBL-TL-PREM(WS-ENTRY-COUNT)
10041             TO PRT-TL-PREM
10042     END-IF
10043     ADD 1 TO WS-ENTRY-COUNT.
10044 7453-EXIT.
10045     EXIT.
10046 *
10047 *
10048 *-----*
10049 * THIS PARAGRAPH WILL PRINT THE SCHEDULE OF AUTOMOBILES. THE *
10050 * PARAGRAPH WILL BE PERFORMED 10 TIMES. THEN THE 7400 PARAGRAPH *
10051 * WILL CONTINUE. IF MORE THAN 10 AUTO EXISTS, THIS PARAGRAPH WILL *
10052 * BE PERFORMED 10 TIMES AGAIN. THIS INFORMATION WILL BE PRINTED *
10053 * ON THE OLD SCHEDULE OF AUTOMOBILIES PAGES. *
10054 *-----*
10055 *
10056 7475-WRITE-AUTO-INFO.
10057 *
10058     MOVE SPACES TO PRINT-LINE.
10059 *
10060     IF WS-TBL-VHDS-AU-NO(WS-ENTRY-COUNT) EQUAL 0
10061         OR WS-TBL-STATE(WS-ENTRY-COUNT) NOT EQUAL WS-HOLD-STATE
10062         MOVE SPACES TO PRINT-LINE
10063     ELSE
10064         MOVE WS-TBL-ID-NO(WS-ENTRY-COUNT) TO PRT-OLD-AUTO-NO
10065         MOVE WS-TBL-YEAR(WS-ENTRY-COUNT)(1:4) TO PRT-OLD-YEAR
10066         STRING WS-TBL-MAKE(WS-ENTRY-COUNT),
10067             " ",
10068             WS-TBL-MODEL(WS-ENTRY-COUNT)
10069             DELIMITED BY " "
10070         INTO PRT-OLD-MAKE-MODEL
10071         MOVE WS-TBL-SERIAL-NO(WS-ENTRY-COUNT)
10072         TO PRT-OLD-SERIAL-NO
10073         IF WS-TBL-CITY(WS-ENTRY-COUNT)(10:10) EQUAL SPACES
10074             STRING WS-TBL-CITY(WS-ENTRY-COUNT)
10075                 DELIMITED BY " "
10076             "("

```

```

10077         DELIMITED BY SIZE
10078         WS-TBL-TERR-CD-OLD(WS-ENTRY-COUNT)
10079         DELIMITED BY " "
10080         ")"
10081         DELIMITED BY SIZE
10082     INTO PRT-OLD-TERR
10083 ELSE
10084     STRING WS-TBL-CITY(WS-ENTRY-COUNT)(1:10),
10085     "("
10086     DELIMITED BY SIZE
10087     WS-TBL-TERR-CD-OLD(WS-ENTRY-COUNT)
10088     DELIMITED BY " "
10089     ")"
10090     DELIMITED BY SIZE
10091     INTO PRT-OLD-TERR
10092 END-IF
10093 if WS-TBL-CLASS-CODE(WS-ENTRY-COUNT) not equal '7960'
10094 if WS-TBL-CLASS-CODE(WS-ENTRY-COUNT) not equal '7961'
10095     if WS-TBL-CLASS-CODE(WS-ENTRY-COUNT) not equal '7963'
10096     IF    WS-TBL-USAGE-TYPE(WS-ENTRY-COUNT) = 'C'
10097         IF WS-TBL-CLASS(WS-ENTRY-COUNT)    = '23'
10098             EVALUATE WS-TBL-CLASS-CODE(WS-ENTRY-COUNT)(3:1)
10099                 WHEN '1'
10100                     MOVE 'LOCAL' TO PRT-OLD-MILEAGE
10101                 WHEN '2'
10102                     MOVE 'INTER' TO PRT-OLD-MILEAGE
10103                 WHEN '3'
10104                     MOVE 'LONG'  TO PRT-OLD-MILEAGE
10105                 WHEN '4'
10106                     MOVE 'LOCAL' TO PRT-OLD-MILEAGE
10107                 WHEN '5'
10108                     MOVE 'INTER' TO PRT-OLD-MILEAGE
10109                 WHEN '6'
10110                     MOVE 'LONG'  TO PRT-OLD-MILEAGE
10111             END-EVALUATE
10112         ELSE
10113             IF WS-TBL-CLASS(WS-ENTRY-COUNT)    = '26'
10114                 MOVE 'LONG' TO PRT-OLD-MILEAGE
10115             ELSE
10116                 MOVE SPACES TO PRT-OLD-MILEAGE
10117             END-IF
10118         END-IF
10119     ELSE
10120         MOVE SPACES TO PRT-OLD-MILEAGE
10121     END-IF
10122 end-if
10123 end-if
10124 end-if
10125     MOVE WS-TBL-USE(WS-ENTRY-COUNT)  TO PRT-OLD-USE
10126     MOVE WS-TBL-WEIGHT(WS-ENTRY-COUNT) TO PRT-OLD-WEIGHT
10127     MOVE WS-TBL-COST(WS-ENTRY-COUNT)  TO PRT-OLD-CAR-COST
10128     MOVE WS-TBL-LOSS-PAYEE(WS-ENTRY-COUNT) TO PRT-LOSS-PAYEE
10129 *
10130 END-IF.
10131 *
10132 WRITE PRINT-LINE AFTER ADVANCING 2 LINES

```

```

10133      ADD 1 TO WS-ENTRY-COUNT.
10134      *
10135      7475-EXIT.
10136      EXIT.
10137      *
10138      *-----*
10139      * THIS PARAGRAPH WILL PRINT THE LOSS PAYEE INFORMATION. *
10140      *-----*
10141      *
10142      7500-PRINT-LOSS-PAYEE.
10143      *
10144      MOVE 1 TO WS-ENTRY-COUNT-4.
10145      MOVE 1 TO WS-LINE-COUNT.
10146      PERFORM 7550-PRINT-LOSS-PAYEE-LINE THRU 7550-EXIT
10147      UNTIL WS-SCH-LP-ACCT(WS-ENTRY-COUNT-4) EQUAL 0.
10148      MOVE 'Y' TO WS-ALRDY-DN.
10149      *
10150      PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
10151      UNTIL WS-LINE-COUNT = 54.
10152      *
10153      MOVE SPACES TO PRINT-LINE.
10154      MOVE WS-COURIER TO PRT-MISC-INFO.
10155      WRITE PRINT-LINE.
10156      *
10157      IF WS-POLY-BSOR-GB-FM-ACCT = 5
10158      MOVE SPACES TO PRINT-LINE
10159      MOVE "State National Insurance Company, Inc."
10160      TO PRT-MISC-7-INFO
10161      WRITE PRINT-LINE
10162      END-IF.
10163      MOVE SPACES TO PRINT-LINE.
10164      MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-7-INFO.
10165      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10166      *
10167      7500-EXIT.
10168      EXIT.
10169      *
10170      *-----*
10171      * THIS PARAGRAPH WILL PRINT THE LOSS PAYEE DETAIL LINE. *
10172      *-----*
10173      *
10174      7550-PRINT-LOSS-PAYEE-LINE.
10175      *
10176      *****-- the code below is to prevent the program from failing due
10177      *****-- to the compute resulting in a zero.
10178
10179      COMPUTE WS-PREV-ENTRY-COUNT = WS-ENTRY-COUNT-4 - 1.
10180      IF WS-PREV-ENTRY-COUNT EQUAL 0
10181      MOVE 500 TO WS-PREV-ENTRY-COUNT
10182      END-IF.
10183      *
10184      IF WS-LINE-COUNT > 47
10185      OR WS-SCH-GARAGE-STATE(WS-ENTRY-COUNT-4) NOT EQUAL
10186      WS-SCH-GARAGE-STATE(WS-PREV-ENTRY-COUNT)
10187      OR WS-SNGL-ST EQUAL 'Y' and WS-ALRDY-DN EQUAL 'N'
10188      IF WS-ENTRY-COUNT-4 NOT EQUAL 1

```

```

10189      PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
10190      UNTIL WS-LINE-COUNT = 54
10191      MOVE SPACES      TO PRINT-LINE
10192      MOVE WS-COURIER  TO PRT-MISC-INFO
10193      WRITE PRINT-LINE
10194      IF WS-POLY-BSOR-GB-FM-ACCT = 5
10195          MOVE SPACES      TO PRINT-LINE
10196          MOVE
10197              "State National Insurance Company, Inc."
10198              to PRT-MISC-7-INFO
10199          WRITE PRINT-LINE
10200      END-IF
10201      MOVE SPACES      TO PRINT-LINE
10202      MOVE WS-PLFD-GB-FORM-NO TO PRT-MISC-7-INFO
10203      WRITE PRINT-LINE AFTER ADVANCING 1 LINES
10204  END-IF
10205
10206  IF WS-SCH-GARAGE-STATE(WS-ENTRY-COUNT-4) NOT EQUAL
10207      WS-SCH-GARAGE-STATE(WS-PREV-ENTRY-COUNT)
10208      MOVE 'AULOSS' TO WS-PLFD-GB-FORM-DESC
10209      MOVE 'Y'      TO WS-PLFD-GB-ENABLE
10210      MOVE 'XX'     TO WS-PLFD-GB-OPTION-1
10211      MOVE 'XX'     TO WS-PLFD-GB-OPTION-2
10212      MOVE WS-SCH-GARAGE-STATE(WS-ENTRY-COUNT-4)
10213          TO WS-PLFD-GB-STATE
10214      PERFORM 8500-GET-MACRO THRU 8500-EXIT
10215          IF SQLCODE EQUAL 1403
10216              MOVE 'XX' TO WS-PLFD-GB-STATE
10217              PERFORM 8500-GET-MACRO THRU 8500-EXIT
10218          END-IF
10219
10220      IF SQLCODE NOT EQUAL 0
10221          MOVE "ERROR SELECTING MACRO FOR STATE XX - 7550"
10222              TO WS-ERR-MSG
10223          PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
10224      END-IF
10225  END-IF
10226
10227      *
10228      MOVE SPACES      TO WS-PRINT-MACRO-LINE
10229      MOVE ESC          TO WS-PRINT-ESC
10230      MOVE WS-PLFD-GB-MACRO TO WS-PRINT-MACRO
10231      MOVE WS-PRINT-MACRO-LINE TO PRINT-LINE
10232      WRITE PRINT-LINE AFTER ADVANCING PAGE
10233      MOVE SPACES      TO PRINT-LINE
10234      MOVE WS-LINEPTR   TO PRT-MISC-INFO
10235      WRITE PRINT-LINE
10236      MOVE 1           TO WS-LINE-COUNT
10237      PERFORM 7700-WRITE-BLANK-LINE THRU 7700-EXIT
10238      VARYING WS-BLANK-LINE FROM 1 BY 1
10239      UNTIL WS-BLANK-LINE > 6
10240      MOVE 7           TO WS-LINE-COUNT
10241  END-IF.
10242
10243      *
10244      MOVE SPACES      TO PRINT-LINE.
10245      MOVE WS-POLY-NO   TO PRT-LP-POLY-NO.

```

```

10245 MOVE WS-SCH-LP-CAR-NO(WS-ENTRY-COUNT-4) TO PRT-LP-AUTO-NO.
10246 MOVE WS-CLIENT-NAME TO PRT-LP-POLY-NAME.
10247 MOVE WS-SCH-LP-NAME(WS-ENTRY-COUNT-4) TO PRT-LP-INFO.
10248 STRING WS-SCH-LP-ORTD(WS-ENTRY-COUNT-4)
10249 DELIMITED BY ''
10250 '#'
10251 DELIMITED BY SIZE
10252 INTO PRT-LP-ORTD
10253 MOVE WS-SCH-LP-ACCT(WS-ENTRY-COUNT-4) TO PRT-LP-ACCT-NO.
10254 WRITE PRINT-LINE AFTER ADVANCING 3 LINES.
10255 ADD 3 TO WS-LINE-COUNT.
10256 *
10257 IF WS-CLIENT-ADNAME NOT EQUAL SPACES
10258 AND WS-SCH-LP-ADNAME(WS-ENTRY-COUNT-4) NOT EQUAL SPACES
10259 MOVE SPACES TO PRINT-LINE
10260 MOVE WS-CLIENT-ADNAME TO PRT-LP-POLY-NAME
10261 MOVE WS-SCH-LP-ADNAME(WS-ENTRY-COUNT-4)
10262 TO PRT-LP-INFO
10263 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
10264 ADD 1 TO WS-LINE-COUNT
10265 END-IF.
10266 *
10267 IF WS-CLIENT-ADNAME EQUAL SPACES
10268 AND WS-SCH-LP-ADNAME(WS-ENTRY-COUNT-4) NOT EQUAL SPACES
10269 MOVE SPACES TO PRINT-LINE
10270 MOVE WS-SCH-LP-ADNAME(WS-ENTRY-COUNT-4)
10271 TO PRT-LP-INFO
10272 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
10273 ADD 1 TO WS-LINE-COUNT
10274 END-IF.
10275 *
10276 IF WS-CLIENT-ADNAME NOT EQUAL SPACES
10277 AND WS-SCH-LP-ADNAME(WS-ENTRY-COUNT-4) EQUAL SPACES
10278 MOVE SPACES TO PRINT-LINE
10279 MOVE WS-CLIENT-ADNAME TO PRT-LP-POLY-NAME
10280 MOVE WS-SCH-LP-STREET1(WS-ENTRY-COUNT-4)
10281 TO PRT-LP-INFO
10282 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
10283 ADD 1 TO WS-LINE-COUNT
10284 ELSE
10285 MOVE SPACES TO PRINT-LINE
10286 MOVE WS-SCH-LP-STREET1(WS-ENTRY-COUNT-4)
10287 TO PRT-LP-INFO
10288 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
10289 ADD 1 TO WS-LINE-COUNT
10290 END-IF.
10291 *
10292 IF WS-SCH-LP-STREET2(WS-ENTRY-COUNT-4) NOT EQUAL SPACES
10293 MOVE SPACES TO PRINT-LINE
10294 MOVE WS-SCH-LP-STREET2(WS-ENTRY-COUNT-4)
10295 TO PRT-LP-INFO
10296 WRITE PRINT-LINE AFTER ADVANCING 1 LINES
10297 ADD 1 TO WS-LINE-COUNT
10298 END-IF.
10299 *
10300 MOVE SPACES TO PRINT-LINE.

```

```

10301      MOVE WS-SCH-LP-CITY-ST-ZIP(WS-ENTRY-COUNT-4)
10302      TO PRT-LP-INFO.
10303      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10304      ADD 1 TO WS-LINE-COUNT.
10305      *
10306      ADD 1 TO WS-ENTRY-COUNT-4.
10307      *
10308      7550-EXIT.
10309      EXIT.
10310      *
10311      *-----*
10312      * THIS PARAGRAPH WILL PRINT THE NOTICE FOR THE LOSS PAYEE.      *
10313      * THIS PARAGRAPH WILL BE PERFORMED FOR EACH LOSS PAYEE.      *
10314      *-----*
10315      *
10316      7600-PRINT-LP-NOTICE.
10317      *
10318      MOVE 1 TO WS-ENTRY-COUNT-3.
10319      *
10320      PERFORM 7650-PRINT-NOTICE THRU 7650-EXIT
10321      VARYING WS-ENTRY-COUNT-3 FROM 1 BY 1
10322      UNTIL WS-LP-NAME(WS-ENTRY-COUNT-3) EQUAL SPACES.
10323      *
10324      7600-EXIT.
10325      EXIT.
10326      *
10327      *-----*
10328      * THIS PARAGRAPH WILL PRINT THE DETAIL LINE FOR THE NOTICE FOR *
10329      * THE LOSS PAYEE.      *
10330      *-----*
10331      *
10332      7650-PRINT-NOTICE.
10333      *
10334      MOVE WS-SV-ENTRY-COUNT(WS-ENTRY-COUNT-3) TO WS-ENTRY-COUNT.
10335      *
10336      MOVE SPACES TO PRINT-LINE.
10337      MOVE WS-REG-FORM-ESC
10338      TO PRT-MISC-INFO.
10339      WRITE PRINT-LINE
10340      AFTER ADVANCING PAGE.
10341      *
10342      MOVE SPACES TO PRINT-LINE.
10343      MOVE WS-BOLD TO PRT-MISC-INFO.
10344      WRITE PRINT-LINE.
10345      *
10346      MOVE SPACES TO PRINT-LINE.
10347      MOVE WS-BOLD TO PRT-MISC-INFO.
10348      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-1
10349      AFTER ADVANCING 1 LINES.
10350      *
10351      MOVE SPACES TO PRINT-LINE.
10352      MOVE WS-12-CHAR TO PRT-MISC-INFO.
10353      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10354      *
10355      MOVE SPACES TO PRINT-LINE.
10356      IF WS-POLY-BSOR-GB-FM-ACCT = 5

```



```

10357         WRITE PRINT-LINE FROM WS-NOTICE-HEAD-SN
10358         AFTER ADVANCING 1 LINES
10359         MOVE "#1 HORTICULTURAL LANE" TO WS-HOLD-COMPANY
10360         MOVE "PO BOX 428"          TO WS-HOLD-POBOX
10361         MOVE "EDWARDSVILLE, ILLINOIS 62025" TO WS-HOLD-CITY
10362     else
10363         WRITE PRINT-LINE FROM WS-NOTICE-HEAD-2
10364         AFTER ADVANCING 1 LINES
10365         MOVE " #1 HORTICULTURAL LANE" TO WS-HOLD-COMPANY
10366         MOVE " PO BOX 428"          TO WS-HOLD-POBOX
10367         MOVE " EDWARDSVILLE, ILLINOIS 62025" TO WS-HOLD-CITY
10368     END-IF.
10369 *
10370     MOVE SPACES      TO PRINT-LINE.
10371     MOVE SPACES      TO WS-NOTICE-HEAD-3.
10372     MOVE WS-CLIENT-NAME TO WS-NOTICE-LP-COMPANY.
10373     MOVE WS-HOLD-COMPANY TO WS-NOTICE-SN-COMPANY.
10374     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10375     AFTER ADVANCING 1 LINES.
10376
10377 * BEGIN PLI31612.4325
10378
10379     MOVE "YES" TO WS-PRINT-CSZ.
10380
10381     IF WS-CLIENT-ADNAME EQUAL SPACES
10382 *****
10383 * IF THERE IS NO DBA NAME, PRINT THE FIRST STREET ADDRESS
10384 *****
10385         MOVE SPACES TO PRINT-LINE
10386         MOVE SPACES TO WS-NOTICE-HEAD-3
10387         MOVE WS-CLIENT-STREET1 TO WS-NOTICE-LP-COMPANY
10388         MOVE WS-HOLD-POBOX TO WS-NOTICE-SN-COMPANY
10389         WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10390         AFTER ADVANCING 1 LINES
10391     ELSE
10392 *****
10393 * IF THERE IS A DBA NAME, PRINT IT BEFORE THE STREET ADDRESS
10394 *****
10395         MOVE SPACES      TO PRINT-LINE
10396         MOVE SPACES      TO WS-NOTICE-HEAD-3
10397         MOVE WS-CLIENT-ADNAME TO WS-NOTICE-LP-COMPANY
10398         MOVE WS-HOLD-POBOX TO WS-NOTICE-SN-COMPANY
10399         WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10400         AFTER ADVANCING 1 LINES
10401         MOVE SPACES TO PRINT-LINE
10402         MOVE SPACES TO WS-NOTICE-HEAD-3
10403         MOVE WS-HOLD-CITY TO WS-NOTICE-SN-COMPANY
10404         MOVE WS-CLIENT-STREET1 TO WS-NOTICE-LP-COMPANY
10405         WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10406         AFTER ADVANCING 1 LINES
10407     END-IF.
10408 *
10409     IF WS-CLIENT-ADNAME EQUAL SPACES
10410         IF WS-CLIENT-STREET2 NOT EQUAL SPACES
10411 *****
10412 * IF THERE IS NO DBA NAME, BUT THERE IS A SECOND STREET ADDRESS

```

```

10413 *****
10414     MOVE SPACES TO PRINT-LINE
10415     MOVE SPACES TO WS-NOTICE-HEAD-3
10416     MOVE WS-CLIENT-STREET2 TO WS-NOTICE-LP-COMPANY
10417     MOVE WS-HOLD-CITY      TO WS-NOTICE-SN-COMPANY
10418     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10419         AFTER ADVANCING 1 LINES
10420     MOVE SPACES TO PRINT-LINE
10421     MOVE SPACES TO WS-NOTICE-HEAD-3
10422     MOVE WS-CLIENT-CITY-ST-ZIP TO WS-NOTICE-LP-COMPANY
10423     MOVE "NO " TO WS-PRINT-CSZ
10424     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10425         AFTER ADVANCING 1 LINES
10426     ELSE
10427 *****
10428 * THERE IS NO DBA NAME, AND THERE IS NO SECOND STREET ADDRESS
10429 *****
10430     MOVE SPACES TO PRINT-LINE
10431     MOVE SPACES TO WS-NOTICE-HEAD-3
10432     MOVE WS-CLIENT-CITY-ST-ZIP TO WS-NOTICE-LP-COMPANY
10433     MOVE WS-HOLD-CITY      TO WS-NOTICE-SN-COMPANY
10434     MOVE "NO " TO WS-PRINT-CSZ
10435     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10436         AFTER ADVANCING 1 LINES
10437     END-IF
10438     ELSE
10439 *****
10440 * THERE IS A DBA NAME, AND THERE IS A SECOND STREET ADDRESS
10441 *****
10442     IF WS-CLIENT-STREET2 NOT EQUAL SPACES
10443     MOVE SPACES TO PRINT-LINE
10444     MOVE SPACES TO WS-NOTICE-HEAD-3
10445     MOVE WS-CLIENT-STREET2 TO WS-NOTICE-LP-COMPANY
10446     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10447         AFTER ADVANCING 1 LINES
10448     MOVE SPACES TO PRINT-LINE
10449     MOVE SPACES TO WS-NOTICE-HEAD-3
10450     MOVE WS-CLIENT-CITY-ST-ZIP TO WS-NOTICE-LP-COMPANY
10451     MOVE "NO " TO WS-PRINT-CSZ
10452     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10453         AFTER ADVANCING 1 LINES
10454     END-IF
10455     END-IF.
10456 *
10457     IF WS-PRINT-CSZ = "YES"
10458     MOVE SPACES TO PRINT-LINE
10459     MOVE SPACES TO WS-NOTICE-HEAD-3
10460     MOVE WS-CLIENT-CITY-ST-ZIP TO WS-NOTICE-LP-COMPANY
10461     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-3
10462         AFTER ADVANCING 1 LINES
10463     END-IF.
10464     MOVE "YES" TO WS-PRINT-CSZ.
10465 *
10466     MOVE SPACES TO PRINT-LINE.
10467     MOVE SPACES TO WS-NOTICE-POLY-NO.
10468     MOVE SPACES TO WS-NOTICE-INFC-EFF.

```

```

10469      MOVE WS-POLY-NO TO WS-NOTICE-POLY-NO.
10470      MOVE WS-INFC-GB-EFF-DATE TO WS-NOTICE-INFC-EFF.
10471      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-7
10472          AFTER ADVANCING 2 LINES.
10473      * END PLI31612.4325
10474
10475      *
10476      MOVE SPACES TO PRINT-LINE.
10477      MOVE SPACES TO WS-NOTICE-POLY-EFF.
10478      MOVE SPACES TO WS-NOTICE-POLY-EXP.
10479      IF WS-POLY-BSOR-GB-FM-ACCT = 5
10480          MOVE WS-POLY-GB-EFF-DATE TO WS-NOTICE-POLY-EFF-SN
10481          MOVE WS-POLY-GB-EXP-DATE TO WS-NOTICE-POLY-EXP-SN
10482          WRITE PRINT-LINE FROM WS-NOTICE-HEAD-8-SN
10483          AFTER ADVANCING 1 LINES
10484      ELSE
10485          MOVE WS-POLY-GB-EFF-DATE TO WS-NOTICE-POLY-EFF
10486          MOVE WS-POLY-GB-EXP-DATE TO WS-NOTICE-POLY-EXP
10487          WRITE PRINT-LINE FROM WS-NOTICE-HEAD-8
10488          AFTER ADVANCING 1 LINES
10489      END-IF.
10490      *
10491      MOVE SPACES TO PRINT-LINE.
10492      MOVE SPACES TO WS-NOTICE-LP-NAME.
10493      MOVE WS-LP-NAME(WS-ENTRY-COUNT-3) TO WS-NOTICE-LP-NAME.
10494      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-9
10495          AFTER ADVANCING 2 LINES.
10496      *
10497      MOVE SPACES TO PRINT-LINE.
10498      MOVE SPACES TO WS-NOTICE-LP-STREET1.
10499      MOVE WS-LP-STREET1(WS-ENTRY-COUNT-3)
10500          TO WS-NOTICE-LP-STREET1.
10501      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-10
10502          AFTER ADVANCING 1 LINES.
10503      *
10504      IF WS-LP-STREET2(WS-ENTRY-COUNT-3) NOT EQUAL SPACES
10505          MOVE SPACES TO PRINT-LINE
10506          MOVE SPACES TO WS-NOTICE-LP-STREET2
10507          MOVE WS-LP-STREET2(WS-ENTRY-COUNT-3)
10508              TO WS-NOTICE-LP-STREET2
10509          WRITE PRINT-LINE FROM WS-NOTICE-HEAD-11
10510              AFTER ADVANCING 1 LINES
10511      END-IF.
10512      *
10513      MOVE SPACES TO PRINT-LINE.
10514      MOVE SPACES TO WS-NOTICE-LP-CITY-ST.
10515      MOVE WS-LP-CITY-ST-ZIP(WS-ENTRY-COUNT-3)
10516          TO WS-NOTICE-LP-CITY-ST.
10517      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-12
10518          AFTER ADVANCING 1 LINES.
10519      *
10520      MOVE SPACES TO PRINT-LINE.
10521      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-13
10522          AFTER ADVANCING 1 LINES.
10523      *
10524      MOVE SPACES TO PRINT-LINE.

```

```

10525      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-13-B
10526      AFTER ADVANCING 1 LINES.
10527  *
10528      MOVE SPACES      TO PRINT-LINE.
10529      MOVE WS-COURIER  TO PRT-MISC-INFO.
10530      WRITE PRINT-LINE.
10531  *
10532      MOVE SPACES TO PRINT-LINE.
10533      MOVE WS-PRT-AUTO-LINE TO PRT-RV-MISC-1-TEXT.
10534      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10535  *
10536  * AUTO DESC
10537  *
10538      MOVE SPACES      TO PRINT-LINE.
10539      MOVE WS-12-CHAR  TO PRT-MISC-INFO.
10540      WRITE PRINT-LINE.
10541  *
10542      MOVE SPACES TO PRINT-LINE.
10543      MOVE "DESCRIPTION OF AUTO(S)" TO PRT-RV-MISC-1-TEXT.
10544      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10545  *
10546      MOVE SPACES TO PRINT-LINE.
10547      STRING WS-TBL-YEAR(WC-ENTRY-COUNT),
10548              " ",
10549              WS-TBL-MAKE(WC-ENTRY-COUNT),
10550              " ",
10551              WS-TBL-MODEL(WC-ENTRY-COUNT)
10552              DELIMITED BY " "
10553      INTO  PRT-RV-MISC-1-TEXT.
10554      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10555  *
10556      MOVE SPACES TO PRINT-LINE.
10557      STRING "VEHICLE SERIAL NO. : ",
10558              WS-TBL-SERIAL-NO(WC-ENTRY-COUNT)
10559              DELIMITED BY SIZE
10560      INTO  PRT-RV-MISC-1-TEXT.
10561      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10562  *
10563      MOVE SPACES      TO PRINT-LINE.
10564      MOVE WS-COURIER  TO PRT-MISC-INFO.
10565      WRITE PRINT-LINE.
10566  *
10567      MOVE SPACES TO PRINT-LINE.
10568      MOVE WS-PRT-AUTO-LINE TO PRT-RV-MISC-1-TEXT.
10569      WRITE PRINT-LINE AFTER ADVANCING 0 LINES.
10570  *
10571      MOVE SPACES      TO PRINT-LINE.
10572      MOVE WS-12-CHAR  TO PRT-MISC-INFO.
10573      WRITE PRINT-LINE.
10574  *
10575      MOVE SPACES TO PRINT-LINE.
10576      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-14
10577      AFTER ADVANCING 1 LINES.
10578  *
10579      MOVE SPACES      TO PRINT-LINE.
10580      MOVE WS-COURIER  TO PRT-MISC-INFO.

```

```

10581      WRITE PRINT-LINE.
10582      *
10583      MOVE SPACES TO PRINT-LINE.
10584      MOVE WS-PRT-AUTO-LINE TO PRT-RV-MISC-1-TEXT.
10585      WRITE PRINT-LINE AFTER ADVANCING 0 LINES.
10586      *
10587      MOVE SPACES TO PRINT-LINE.
10588      MOVE WS-12-CHAR TO PRT-MISC-INFO.
10589      WRITE PRINT-LINE.
10590      *
10591      * LIABILITY
10592      *
10593      MOVE SPACES TO PRINT-LINE.
10594      MOVE SPACES TO WS-NOTICE-HEAD-15.
10595      MOVE "LIABILITY" TO WS-NOTICE-COV-TYPE.
10596      *
10597      IF WS-LP-LIMIT-SPLIT-HOLD(WS-ENTRY-COUNT-3) = 'Y'
10598          MOVE WS-LP-LIMIT-HOLD(WS-ENTRY-COUNT-3) TO
10599              WS-AUCV-AU-SPLIT-LIMIT
10600          PERFORM 7680-FORMAT-SPLIT-LIMIT-3 THRU 7680-EXIT
10601          MOVE WS-AUCV-AU-SPLIT-LIMIT-3 TO WS-NOTICE-ACV-TXT
10602          MOVE SPACES TO WS-NOTICE-LIMIT-EACH
10603      ELSE
10604          IF WS-TBL-LIAB-LT(WS-ENTRY-COUNT) = "SEE END"
10605              PERFORM 7675-STRING-LIABILITY-AMOUNTS THRU 7675-EXIT
10606          ELSE
10607              IF WS-TBL-LIAB-LT(WS-ENTRY-COUNT) NUMERIC
10608                  MOVE WS-TBL-LIAB-LT(WS-ENTRY-COUNT)
10609                      TO WS-NOTICE-LIMIT-AMT
10610              ELSE
10611                  MOVE WS-TBL-LIAB-LT(WS-ENTRY-COUNT)
10612                      TO WS-NOTICE-LIMIT-AMT-CHAR
10613              END-IF
10614          END-IF
10615          MOVE " EACH ACCIDENT" TO WS-NOTICE-LIMIT-EACH
10616      END-IF.
10617      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-15
10618          AFTER ADVANCING 1 LINES.
10619      *
10620      * PIP
10621      *
10622      MOVE SPACES TO PRINT-LINE.
10623      MOVE SPACES TO WS-NOTICE-HEAD-15.
10624      MOVE "PERSONAL INJURY PROTECTION" TO WS-NOTICE-COV-TYPE.
10625      IF WS-PIP-EXIST(WS-ENTRY-COUNT) = 'Y'
10626          MOVE "NO DEDUCTIBLE" TO WS-NOTICE-ACV-TXT
10627      ELSE
10628          MOVE "NOT APPLICABLE" TO WS-NOTICE-ACV-TXT
10629      END-IF.
10630      MOVE SPACES TO PRINT-LINE.
10631      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-15
10632          AFTER ADVANCING 2 LINES.
10633      *
10634      * EQUIVALENT
10635      *
10636      MOVE SPACES TO PRINT-LINE.

```

```

10637      MOVE SPACES TO WS-NOTICE-HEAD-15.
10638      MOVE " (OR EQUIVALENT NO-FAULT COVERAGE)"
10639      TO WS-NOTICE-COV-TYPE.
10640      MOVE SPACES TO PRINT-LINE.
10641      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-15
10642          AFTER ADVANCING 1 LINES.
10643      *
10644      * COMP
10645      *
10646      MOVE SPACES TO PRINT-LINE.
10647      MOVE SPACES TO WS-NOTICE-HEAD-15.
10648      MOVE "COMPREHENSIVE" TO WS-NOTICE-COV-TYPE.
10649      MOVE "ACV OR COST OF REPAIR WHICHEVER IS LESS; MINUS;"
10650      TO WS-NOTICE-ACV-TXT.
10651      MOVE SPACES TO PRINT-LINE.
10652      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-15
10653          AFTER ADVANCING 2 LINES.
10654      *
10655      MOVE SPACES TO PRINT-LINE.
10656      MOVE SPACES TO WS-NOTICE-HEAD-15.
10657      MOVE WS-TBL-COMP-DEDUCT(WS-ENTRY-COUNT)
10658      TO WS-NOTICE-DEDUCT-AMT.
10659      MOVE "FOR EACH COVERED AUTO" TO WS-NOTICE-DEDUCT-COV.
10660      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-15
10661          AFTER ADVANCING 1 LINES.
10662      *
10663      * COLL
10664      *
10665      MOVE SPACES TO PRINT-LINE.
10666      MOVE SPACES TO WS-NOTICE-HEAD-15.
10667      MOVE "COLLISION" TO WS-NOTICE-COV-TYPE.
10668      MOVE "ACV OR COST OF REPAIR WHICHEVER IS LESS; MINUS;"
10669      TO WS-NOTICE-ACV-TXT.
10670      MOVE SPACES TO PRINT-LINE.
10671      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-15
10672          AFTER ADVANCING 2 LINES.
10673      *
10674      MOVE SPACES TO PRINT-LINE.
10675      MOVE SPACES TO WS-NOTICE-HEAD-15.
10676      IF WS-LIMF(WS-ENTRY-COUNT) = 'N'
10677          MOVE WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
10678          TO WS-NOTICE-DEDUCT-AMT
10679      ELSE
10680          MOVE WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
10681          TO WS-ZERO-NOT-DEDUCT-AMT
10682      END-IF.
10683      MOVE "FOR EACH COVERED AUTO" TO WS-NOTICE-DEDUCT-COV.
10684      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-15
10685          AFTER ADVANCING 1 LINES.
10686      MOVE SPACES TO WS-NOTICE-HEAD-15.
10687      *
10688      MOVE SPACES TO PRINT-LINE.
10689      MOVE WS-COURIER TO PRT-MISC-INFO.
10690      WRITE PRINT-LINE.
10691      *
10692      MOVE SPACES TO PRINT-LINE.

```



```

10693      MOVE WS-PRT-AUTO-LINE TO PRT-RV-MISC-1-TEXT.
10694      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10695      *
10696      MOVE SPACES      TO PRINT-LINE.
10697      MOVE WS-12-CHAR  TO PRT-MISC-INFO.
10698      WRITE PRINT-LINE.
10699      *
10700      MOVE SPACES TO PRINT-LINE.
10701      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-16
10702          AFTER ADVANCING 1 LINES.
10703      *
10704      MOVE SPACES      TO PRINT-LINE.
10705      MOVE WS-LINEPTR  TO PRT-MISC-INFO.
10706      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10707      *
10708      IF WS-LP-EXP-DATE(WS-ENTRY-COUNT-3) EQUAL SPACES
10709          MOVE 'X' TO WS-NOTICE-HEAD-17-X
10710      ELSE
10711          MOVE SPACES TO WS-NOTICE-HEAD-17-X
10712      END-IF.
10713      *
10714      MOVE SPACES TO PRINT-LINE.
10715      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-17
10716          AFTER ADVANCING 1 LINES.
10717      *
10718      MOVE SPACES TO PRINT-LINE.
10719      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-18
10720          AFTER ADVANCING 1 LINES.
10721      *
10722      MOVE SPACES TO PRINT-LINE.
10723      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-19
10724          AFTER ADVANCING 1 LINES.
10725      *
10726      MOVE SPACES TO PRINT-LINE.
10727      WRITE PRINT-LINE
10728          AFTER ADVANCING 1 LINES.
10729      *
10730      MOVE SPACES TO PRINT-LINE.
10731      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-20
10732          AFTER ADVANCING 1 LINES.
10733      *
10734      MOVE SPACES TO PRINT-LINE.
10735      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-21
10736          AFTER ADVANCING 1 LINES.
10737      *
10738      MOVE SPACES TO PRINT-LINE.
10739      WRITE PRINT-LINE
10740          AFTER ADVANCING 1 LINES.
10741      *
10742      IF WS-LP-EXP-DATE(WS-ENTRY-COUNT-3) NOT EQUAL SPACES
10743          MOVE 'X' TO WS-NOTICE-HEAD-22-X
10744      ELSE
10745          MOVE SPACES TO WS-NOTICE-HEAD-22-X
10746      END-IF.
10747      MOVE SPACES TO PRINT-LINE.
10748      WRITE PRINT-LINE FROM WS-NOTICE-HEAD-22

```

```

10749         AFTER ADVANCING 1 LINES.
10750     *
10751     MOVE SPACES TO PRINT-LINE.
10752     WRITE PRINT-LINE FROM WS-NOTICE-HEAD-23
10753         AFTER ADVANCING 1 LINES.
10754     *
10755     IF WS-LP-EXP-DATE(WS-ENTRY-COUNT-3) NOT EQUAL SPACES
10756         MOVE SPACES TO PRINT-LINE
10757         MOVE SPACES TO WS-NOTICE-HEAD-24-DATE
10758         MOVE WS-LP-EXP-DATE(WS-ENTRY-COUNT-3)
10759             TO WS-NOTICE-HEAD-24-DATE
10760         WRITE PRINT-LINE FROM WS-NOTICE-HEAD-24
10761             AFTER ADVANCING 0 LINES
10762     END-IF.
10763     *
10764
10765     if WS-POLY-BSOR-GB-FM-ACCT = 5
10766         MOVE SPACES TO PRINT-LINE
10767         MOVE "State National Insurance Company, Inc."
10768             TO PRT-RV-MISC-1-TEXT
10769         WRITE PRINT-LINE AFTER ADVANCING 3 LINES
10770         MOVE "AMA-104-SN (09/12)" TO WS-HOLD-EDITION
10771     else
10772         MOVE SPACES TO PRINT-LINE
10773         MOVE "FLORISTS' MUTUAL INSURANCE COMPANY"
10774             TO PRT-RV-MISC-1-TEXT
10775         WRITE PRINT-LINE AFTER ADVANCING 3 LINES
10776         MOVE "AMA-104(7/85)" TO WS-HOLD-EDITION
10777     end-if.
10778     *
10779     MOVE SPACES TO PRINT-LINE.
10780     IF WS-CLIENT-STATE = "HI" THEN
10781         MOVE "AMA-104(7/91)"
10782             TO PRT-RV-MISC-1-TEXT
10783     ELSE
10784         MOVE WS-HOLD-EDITION
10785             TO PRT-RV-MISC-1-TEXT
10786     END-IF.
10787     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10788     *
10789     MOVE SPACES TO PRINT-LINE.
10790     MOVE WS-12-CHAR TO PRT-MISC-INFO.
10791     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
10792     *
10793     7650-EXIT.
10794     EXIT.
10795     *
10796     7675-STRING-LIABILITY-AMOUNTS.
10797     *
10798     STRING WS-TBL-PD-LT(WS-ENTRY-COUNT) DELIMITED BY " "
10799         INTO WS-PD-AMT.
10800     *
10801     PERFORM VARYING WS-POSITION-COUNT FROM 1 BY 1
10802         UNTIL WS-TEST-POSITION EQUAL "/"
10803         OR WS-TEST-POSITION EQUAL SPACES
10804         MOVE WS-PD-AMT(WS-POSITION-COUNT:1)

```

```

10805         TO WS-TEST-POSITION
10806         MOVE WS-POSITION-COUNT TO WS-PD-AMT-LENGTH
10807     END-PERFORM.
10808 *
10809     COMPUTE WS-PD-AMT-LENGTH = (WS-PD-AMT-LENGTH - 1).
10810     PERFORM VARYING WS-PD-AMT-LENGTH FROM WS-PD-AMT-LENGTH BY -1
10811         UNTIL WS-PD-AMT-LENGTH = 0
10812         COMPUTE WS-COMMA-POSITION-COUNT =
10813             WS-COMMA-POSITION-COUNT + 1
10814         IF WS-COMMA-POSITION-COUNT = 4 OR
10815             WS-COMMA-POSITION-COUNT = 7
10816             STRING WS-PD-AMT(WS-PD-AMT-LENGTH:1),
10817                 WS-COMMA,
10818                 WS-PD-OUT-HOLD
10819                 DELIMITED BY SIZE
10820             INTO WS-PD-OUT
10821         ELSE
10822             STRING WS-PD-AMT(WS-PD-AMT-LENGTH:1),
10823                 WS-PD-OUT-HOLD
10824                 DELIMITED BY SIZE
10825             INTO WS-PD-OUT
10826         END-IF
10827         MOVE WS-PD-OUT TO WS-PD-OUT-HOLD
10828     END-PERFORM.
10829     STRING WS-DOLLAR-SIGN,
10830         WS-PD-OUT
10831         DELIMITED BY " "
10832     INTO WS-PD-OUT-FINAL.
10833 *
10834     MOVE LOW-VALUES TO WS-TEST-POSITION.
10835     MOVE 0 TO WS-COMMA-POSITION-COUNT.
10836     MOVE SPACES TO WS-BI-OUT-HOLD.
10837     MOVE 0 TO WS-BI-AMT-LENGTH.
10838 *
10839     PERFORM VARYING WS-POSITION-COUNT FROM 1 BY 1
10840         UNTIL WS-TEST-POSITION EQUAL "/"
10841         OR WS-TEST-POSITION EQUAL SPACES
10842         MOVE WS-TBL-BI-LT(WS-ENTRY-COUNT)
10843         (WS-POSITION-COUNT:1) TO WS-TEST-POSITION
10844         MOVE WS-POSITION-COUNT TO WS-POSITION-COUNT-HOLD
10845     END-PERFORM.
10846     MOVE WS-TBL-BI-LT(WS-ENTRY-COUNT) (1:WS-POSITION-COUNT - 2)
10847     TO WS-BI-AMT.
10848     COMPUTE WS-BI-AMT-LENGTH = (WS-POSITION-COUNT - 2).
10849 *
10850     PERFORM VARYING WS-BI-AMT-LENGTH FROM WS-BI-AMT-LENGTH BY -1
10851         UNTIL WS-BI-AMT-LENGTH = 0
10852         COMPUTE WS-COMMA-POSITION-COUNT =
10853             WS-COMMA-POSITION-COUNT + 1
10854         IF WS-COMMA-POSITION-COUNT = 4 OR
10855             WS-COMMA-POSITION-COUNT = 7
10856             STRING WS-BI-AMT(WS-BI-AMT-LENGTH:1),
10857                 WS-COMMA,
10858                 WS-BI-OUT-HOLD
10859                 DELIMITED BY SIZE
10860             INTO WS-BI-OUT

```

```

10861         ELSE
10862             STRING WS-BI-AMT(WS-BI-AMT-LENGTH:1),
10863                 WS-BI-OUT-HOLD
10864                 DELIMITED BY SIZE
10865             INTO WS-BI-OUT
10866         END-IF
10867         MOVE WS-BI-OUT TO WS-BI-OUT-HOLD
10868     END-PERFORM.
10869     STRING WS-DOLLAR-SIGN,
10870         WS-BI-OUT
10871         DELIMITED BY " "
10872     INTO WS-BI-OUT-FINAL.
10873 *
10874     PERFORM VARYING WS-POSITION-COUNT
10875     FROM WS-POSITION-COUNT BY 1
10876     UNTIL WS-TEST-POSITION EQUAL " "
10877         MOVE WS-TBL-BI-LT(WS-ENTRY-COUNT)
10878         (WS-POSITION-COUNT:1) TO WS-TEST-POSITION
10879     END-PERFORM.
10880     COMPUTE WS-POSITION-COUNT = WS-POSITION-COUNT - 2.
10881     MOVE WS-TBL-BI-LT(WS-ENTRY-COUNT)
10882         (WS-POSITION-COUNT-HOLD + 1:
10883         WS-POSITION-COUNT - WS-POSITION-COUNT-HOLD)
10884     TO WS-BI-AMT-2.
10885     MOVE LOW-VALUES TO WS-TEST-POSITION.
10886     MOVE 0 TO WS-COMMA-POSITION-COUNT.
10887     MOVE SPACES TO WS-BI-OUT-HOLD.
10888     MOVE 0 TO WS-BI-AMT-LENGTH.
10889 *
10890     COMPUTE WS-BI-AMT-LENGTH = (WS-POSITION-COUNT -
10891         WS-POSITION-COUNT-HOLD).
10892     PERFORM VARYING WS-BI-AMT-LENGTH FROM WS-BI-AMT-LENGTH BY -1
10893     UNTIL WS-BI-AMT-LENGTH = 0
10894         COMPUTE WS-COMMA-POSITION-COUNT =
10895             WS-COMMA-POSITION-COUNT + 1
10896         IF WS-COMMA-POSITION-COUNT = 4 OR
10897             WS-COMMA-POSITION-COUNT = 7
10898             STRING WS-BI-AMT-2(WS-BI-AMT-LENGTH:1),
10899                 WS-COMMA,
10900                 WS-BI-OUT-HOLD
10901                 DELIMITED BY SIZE
10902             INTO WS-BI-OUT-2
10903         ELSE
10904             STRING WS-BI-AMT-2(WS-BI-AMT-LENGTH:1),
10905                 WS-BI-OUT-HOLD
10906                 DELIMITED BY SIZE
10907             INTO WS-BI-OUT-2
10908         END-IF
10909         MOVE WS-BI-OUT-2 TO WS-BI-OUT-HOLD
10910     END-PERFORM.
10911     STRING WS-DOLLAR-SIGN,
10912         WS-BI-OUT-2
10913         DELIMITED BY " "
10914     INTO WS-BI-OUT-2-FINAL.
10915 *
10916     STRING WS-BI-OUT-FINAL,

```

```

10917         WS-SLASH,
10918         WS-BI-OUT-2-FINAL,
10919         WS-SLASH,
10920         WS-PD-OUT-FINAL
10921         DELIMITED BY " "
10922     INTO WS-NOTICE-ACV-TXT.
10923 *
10924     7675-EXIT.
10925     EXIT.
10926 *
10927     7680-FORMAT-SPLIT-LIMIT-3.
10928 *
10929 * THIS PARAGRAPH WILL TAKE A SPLIT LIMIT OF 500000/1000000 AND FORMAT THE
OUTPUT AS
10930 * $500,000/$1,000,000/$500,000. USED FOR SPL AND SPLPD COVERAGE.
10931 *
10932     EXEC SQL AT ORA1
10933     SELECT
10934         LTRIM(TO_CHAR(TO_NUMBER(SUBSTR(:WS-AUCV-AU-SPLIT-LIMIT,
10935 -         1,INSTR(:WS-AUCV-AU-SPLIT-LIMIT,'/)-1)),$9,999,999')
10936 -         ||'/'||
10937 -         LTRIM(TO_CHAR(TO_NUMBER(SUBSTR(:WS-AUCV-AU-SPLIT-LIMIT,
10938 (INSTR(:WS-AUCV-AU-SPLIT-LIMIT,'/')+1),
10939 (LENGTH(:WS-AUCV-AU-SPLIT-LIMIT)))),$9,999,999'),' ')||
10940         '/'||LTRIM(TO_CHAR(TO_NUMBER(:WS-PD-LIMIT-HOLD),
10941 -         '$9,999,999'),' '),',' ' )
10942     INTO :WS-AUCV-AU-SPLIT-LIMIT-3
10943     FROM DUAL
10944     END-EXEC.
10945 *
10946     7680-EXIT.
10947     EXIT.
10948 *
10949     7685-FORMAT-SPLIT-LIMIT-2.
10950 *
10951 * THIS PARAGRAPH WILL TAKE A SPLIT LIMIT OF 500000/1000000 AND FORMAT THE
OUTPUT AS
10952 * $500,000/$1,000,000. USED FOR UM AND UI COVERAGE.
10953 *
10954     EXEC SQL AT ORA1
10955     SELECT
10956         LTRIM(TO_CHAR(TO_NUMBER(SUBSTR(:WS-AUCV-AU-SPLIT-LIMIT-2,
10957 -         1,INSTR(:WS-AUCV-AU-SPLIT-LIMIT-2,'/)-1)),$9,999,999')
10958 -         ||'/'||
10959 -         LTRIM(TO_CHAR(TO_NUMBER(SUBSTR(:WS-AUCV-AU-SPLIT-LIMIT-2,
10960 (INSTR(:WS-AUCV-AU-SPLIT-LIMIT-2,'/')+1),
10961 (LENGTH(:WS-AUCV-AU-SPLIT-LIMIT-2)))),$9,999,999'),' '),',' ' )
10962 -         '$9,999,999'),' '),',' ' )
10963     INTO :WS-AUCV-AU-SPLIT-LIMIT-2
10964     FROM DUAL
10965     END-EXEC.
10966 *
10967     7685-EXIT.
10968     EXIT.
10969 *-----*
10970 * THIS PARAGRAPH WILL PRINT BLANK LINES.

```

10971	* ----- *	
10972	*	
10973	7700-WRITE-BLANK-LINE.	
10974	*	
10975	MOVE SPACES TO PRINT-LINE.	
10976	WRITE PRINT-LINE AFTER ADVANCING 1 LINES.	
10977	*	
10978	ADD 1 TO WS-LINE-COUNT.	
10979	*	
10980	7700-EXIT.	
10981	EXIT.	
10982	*	
10983	* ----- *	
10984	* THIS PARAGRAPH WILL PRINT THE REVISION PAGE.	*
10985	* ----- *	
10986	*	
10987	7800-PRINT-RV-POLICY.	
10988		
10989		
10990		
10991	display 'printing REVISION page'.	
10992		
10993		
10994	*	
10995	MOVE SPACES TO PRINT-LINE.	
10996	MOVE WS-REG-FORM-ESC	
10997	TO PRT-MISC-INFO.	
10998	WRITE PRINT-LINE	
10999	AFTER ADVANCING PAGE.	
11000	*	
11001	MOVE SPACES TO PRINT-LINE.	
11002	MOVE WS-BOLD TO PRT-MISC-INFO.	
11003	WRITE PRINT-LINE.	
11004	*	
11005	MOVE SPACES TO PRINT-LINE.	
11006	WRITE PRINT-LINE FROM WS-PRT-RV-HEADING	
11007	AFTER ADVANCING 1 LINES.	
11008	*	
11009	MOVE SPACES TO PRINT-LINE.	
11010	MOVE WS-INFC-GB-END-NO TO WS-PRT-RV-AUTO-END-NO.	
11011	WRITE PRINT-LINE FROM WS-PRT-RV-AUTO-END	
11012	AFTER ADVANCING 1 LINES.	
11013	*	
11014	MOVE SPACES TO PRINT-LINE.	
11015	MOVE WS-LINEPTR TO PRT-MISC-INFO.	
11016	WRITE PRINT-LINE AFTER ADVANCING 1 LINES.	
11017	*	
11018	MOVE SPACES TO PRINT-LINE.	
11019	MOVE WS-POLY-NO TO WS-PRT-RV-POLY-NO.	
11020	MOVE WS-POLY-BSOR-GB-ACCOUNT TO WS-PRT-RV-ACCT-NO.	
11021	WRITE PRINT-LINE FROM WS-PRT-RV-AUTO-POLY	
11022	AFTER ADVANCING 2 LINES.	
11023	*	
11024	if WS-POLY-BSOR-GB-FM-ACCT = 5	
11025	MOVE SPACES TO PRINT-LINE	
11026	MOVE WS-SN-111-ISSUED	


```

11027         TO PRT-RV-MISC-1-TEXT
11028     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
11029 else
11030     MOVE SPACES TO PRINT-LINE
11031     MOVE " FLORISTS' MUTUAL INSURANCE COMPANY ISSUED TO:"
11032     TO PRT-RV-MISC-1-TEXT
11033     WRITE PRINT-LINE AFTER ADVANCING 1 LINES
11034 end-if.
11035 *
11036     MOVE SPACES TO PRINT-LINE.
11037     MOVE WS-COURIER TO PRT-MISC-INFO.
11038     WRITE PRINT-LINE AFTER ADVANCING 1 LINE.
11039 *
11040     MOVE SPACES TO PRINT-LINE.
11041     MOVE WS-CLIENT-NAME TO PRT-RV-MISC-1-TEXT.
11042     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11043 *
11044     MOVE SPACES TO PRINT-LINE.
11045     IF WS-CLIENT-ADNAME NOT EQUAL SPACES
11046         MOVE WS-CLIENT-ADNAME TO PRT-RV-MISC-1-TEXT
11047         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
11048     END-IF.
11049 *
11050     MOVE SPACES TO PRINT-LINE.
11051     MOVE WS-CLIENT-STREET1 TO PRT-RV-MISC-1-TEXT.
11052     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11053 *
11054     IF WS-CLIENT-STREET2 NOT EQUAL SPACES
11055         MOVE SPACES TO PRINT-LINE
11056         MOVE WS-CLIENT-STREET2 TO PRT-RV-MISC-1-TEXT
11057         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
11058     END-IF.
11059 *
11060     MOVE SPACES TO PRINT-LINE.
11061     IF WS-CLIENT-ZIP(6:4) = SPACES
11062         STRING WS-CLIENT-CITY DELIMITED BY " "
11063             " ",
11064             WS-CLIENT-STATE,
11065             " ",
11066             WS-CLIENT-ZIP(1:5)
11067             DELIMITED BY SIZE
11068         INTO PRT-RV-MISC-1-TEXT
11069     ELSE
11070         STRING WS-CLIENT-CITY DELIMITED BY " "
11071             " ",
11072             WS-CLIENT-STATE,
11073             " ",
11074             WS-CLIENT-ZIP(1:5),
11075             "-",
11076             WS-CLIENT-ZIP(6:4)
11077             DELIMITED BY SIZE
11078         INTO PRT-RV-MISC-1-TEXT
11079     END-IF.
11080     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11081 *
11082     MOVE SPACES TO PRINT-LINE.

```

```

11083      MOVE WS-INFC-GB-EFF-DATE TO WS-PRT-RV-INFC-EFF.
11084      WRITE PRINT-LINE FROM WS-PRT-RV-INFC-DATE
11085          AFTER ADVANCING 2 LINES.
11086  *
11087      MOVE SPACES          TO PRINT-LINE.
11088      MOVE WS-POLY-GB-EFF-DATE TO WS-PRT-RV-POLY-EFF.
11089      MOVE WS-POLY-GB-EXP-DATE TO WS-PRT-RV-POLY-EXP.
11090      WRITE PRINT-LINE FROM WS-PRT-RV-POLY-DATE
11091          AFTER ADVANCING 2 LINES.
11092  *
11093      MOVE SPACES TO PRINT-LINE.
11094      IF WS-INFC-GB-WRITTEN-PX >= 0
11095          MOVE WS-PRT-RV-VEHL-TEXT-ADD TO WS-PRT-RV-VEHL-TEXT
11096      ELSE
11097          MOVE WS-PRT-RV-VEHL-TEXT-RETURN TO WS-PRT-RV-VEHL-TEXT
11098      END-IF.
11099      IF WS-ENTRY-COUNT = 1
11100          MOVE '$'          TO WS-PRT-RV-VEHL-PX-DOL
11101          MOVE WS-INFC-GB-WRITTEN-PX TO WS-PRT-RV-VEHL-PX
11102      ELSE
11103          MOVE ':'          TO WS-PRT-RV-VEHL-PX-DOL
11104          MOVE "INCLUDED"   TO WS-PRT-RV-VEHL-INCL
11105      END-IF.
11106      WRITE PRINT-LINE FROM WS-PRT-RV-VEHL-PX-LINE
11107          AFTER ADVANCING 2 LINES.
11108  *
11109      MOVE SPACES TO PRINT-LINE.
11110      EVALUATE TRUE
11111          WHEN WS-TBL-ADDED-EXIT(WS-ENTRY-COUNT)
11112              = WS-TBL-TL-EXIT(WS-ENTRY-COUNT)
11113              MOVE WS-PRT-RV-AUTO-ADD TO PRT-RV-MISC-1-TEXT
11114          WHEN WS-TBL-DROPPED-EXIT(WS-ENTRY-COUNT)
11115              = WS-TBL-TL-EXIT(WS-ENTRY-COUNT)
11116              MOVE WS-PRT-RV-AUTO-DEL TO PRT-RV-MISC-1-TEXT
11117          WHEN OTHER
11118              MOVE WS-PRT-RV-AUTO-AMENDED TO PRT-RV-MISC-1-TEXT
11119      END-EVALUATE.
11120      WRITE PRINT-LINE AFTER ADVANCING 2 LINES.
11121  *
11122      MOVE SPACES TO PRINT-LINE.
11123      MOVE WS-PRT-AUTO-LINE TO PRT-RV-MISC-1-TEXT.
11124      WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11125  *
11126  * CAR INFO
11127  *
11128      MOVE SPACES          TO PRINT-LINE.
11129      MOVE SPACES          TO WS-PRT-RV-AUTO.
11130      MOVE "NO:"           TO WS-PRT-RV-AUTO-DESC-1.
11131      MOVE WS-TBL-ID-NO(WS-ENTRY-COUNT) TO WS-PRT-RV-AUTO-VAR-1.
11132      WRITE PRINT-LINE FROM WS-PRT-RV-AUTO
11133          AFTER ADVANCING 2 LINES.
11134  *
11135      MOVE SPACES TO PRINT-LINE.
11136      MOVE SPACES TO WS-PRT-RV-AUTO.
11137      MOVE "DESC." TO WS-PRT-RV-AUTO-DESC-1.
11138      MOVE ":"     TO WS-PRT-HYP-1.

```

```

11139 STRING
11140     WS-TBL-YEAR(WS-ENTRY-COUNT),
11141     " ",
11142     " ",
11143     WS-TBL-MAKE(WS-ENTRY-COUNT),
11144     " ",
11145     " ",
11146     WS-TBL-MODEL(WS-ENTRY-COUNT)
11147     DELIMITED BY " "
11148 INTO WS-PRT-RV-AUTO-VAR-1.
11149 MOVE "ID NO."          TO WS-PRT-RV-AUTO-DESC-2.
11150 MOVE ":"              TO WS-PRT-HYP-2.
11151 MOVE WS-TBL-SERIAL-NO(WS-ENTRY-COUNT)
11152 TO WS-PRT-RV-AUTO-VAR-2-TXT.
11153 WRITE PRINT-LINE FROM WS-PRT-RV-AUTO
11154     AFTER ADVANCING 2 LINES.
11155 *
11156 MOVE SPACES TO PRINT-LINE.
11157 MOVE SPACES TO WS-PRT-RV-AUTO.
11158 MOVE "GARAGED" TO WS-PRT-RV-AUTO-DESC-1.
11159 MOVE ":" TO WS-PRT-HYP-1.
11160 STRING WS-TBL-CITY(WS-ENTRY-COUNT)
11161     DELIMITED BY " "
11162     " ",
11163     WS-TBL-STATE(WS-ENTRY-COUNT)
11164     DELIMITED BY SIZE
11165 INTO WS-PRT-RV-AUTO-VAR-1.
11166 MOVE "CLASS" TO WS-PRT-RV-AUTO-DESC-2.
11167 MOVE ":" TO WS-PRT-HYP-2.
11168 MOVE WS-TBL-CLASS-CODE(WS-ENTRY-COUNT)
11169 TO WS-PRT-RV-AUTO-VAR-2-TXT.
11170 WRITE PRINT-LINE FROM WS-PRT-RV-AUTO
11171     AFTER ADVANCING 1 LINES.
11172 *
11173 MOVE SPACES TO PRINT-LINE.
11174 MOVE SPACES TO WS-PRT-RV-AUTO.
11175 MOVE "TERR." TO WS-PRT-RV-AUTO-DESC-1.
11176 MOVE ":" TO WS-PRT-HYP-1.
11177 MOVE WS-TBL-TERR-CD-OLD(WS-ENTRY-COUNT)
11178 TO WS-PRT-RV-AUTO-VAR-1.
11179 MOVE "COST NEW" TO WS-PRT-RV-AUTO-DESC-2.
11180 MOVE ":" TO WS-PRT-HYP-2.
11181 MOVE WS-TBL-COST(WS-ENTRY-COUNT)
11182 TO WS-PRT-RV-AUTO-VAR-2-AMT.
11183 WRITE PRINT-LINE FROM WS-PRT-RV-AUTO
11184     AFTER ADVANCING 1 LINES.
11185 *
11186 MOVE SPACES TO PRINT-LINE.
11187 WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11188 *
11189 MOVE SPACES TO PRINT-LINE.
11190 MOVE SPACES TO WS-PRT-AUTO-COV-HEAD-1-TXT.
11191 IF WS-TBL-DROPPED-EXIT(WS-ENTRY-COUNT)
11192     = WS-TBL-TL-EXIT(WS-ENTRY-COUNT)
11193     AND WS-TBL-TL-PREM(WS-ENTRY-COUNT) = 0
11194     MOVE " RETURN" TO WS-PRT-AUTO-COV-HEAD-1-TXT

```

```

11195 ELSE
11196     IF WS-TBL-TL-PREM(WS-ENTRY-COUNT) >= 0
11197         MOVE "ADDITIONAL" TO WS-PRT-AUTO-COV-HEAD-1-TXT
11198     ELSE
11199         MOVE " RETURN" TO WS-PRT-AUTO-COV-HEAD-1-TXT
11200     END-IF
11201 END-IF.
11202 WRITE PRINT-LINE FROM WS-PRT-AUTO-COV-HEAD-1
11203     AFTER ADVANCING 1 LINES.
11204 *
11205 MOVE SPACES TO PRINT-LINE.
11206 WRITE PRINT-LINE FROM WS-PRT-AUTO-COV-HEAD-2
11207     AFTER ADVANCING 1 LINES.
11208 *
11209 MOVE SPACES TO PRINT-LINE.
11210 WRITE PRINT-LINE FROM WS-PRT-AUTO-COV-HEAD-3
11211     AFTER ADVANCING 1 LINES.
11212 *
11213 * PRINT LIABILITY LINE
11214 *
11215 MOVE SPACES TO PRINT-LINE.
11216 MOVE SPACES TO WS-PRT-RV-COV.
11217 MOVE WS-TBL-LIAB-WRIT(WS-ENTRY-COUNT) TO WS-LIAB-COST.
11218 ADD WS-TBL-PD-WRIT(WS-ENTRY-COUNT) TO WS-LIAB-COST.
11219 ADD WS-TBL-BI-WRIT(WS-ENTRY-COUNT) TO WS-LIAB-COST.
11220 *
11221 MOVE WS-TBL-LIAB-NET(WS-ENTRY-COUNT) TO WS-LIAB-COST-NET.
11222 ADD WS-TBL-PD-NET(WS-ENTRY-COUNT) TO WS-LIAB-COST-NET.
11223 ADD WS-TBL-BI-NET(WS-ENTRY-COUNT) TO WS-LIAB-COST-NET.
11224 *
11225 MOVE "LIABILITY" TO WS-PRT-RV-COV-DESC.
11226 MOVE "$" TO WS-PRT-RV-COV-DOLLAR.
11227 MOVE "$" TO WS-PRT-RV-COV-DOLLAR-1.
11228 *
11229 IF WS-SLL-EXIST(WS-ENTRY-COUNT) = 'Y'
11230     IF WS-TBL-LIAB-LT(WS-ENTRY-COUNT) = "SEE END"
11231         IF WS-AUCV-AU-LIMIT-SPLIT-BI = 'Y'
11232             MOVE "SEE POLICY" TO WS-PRT-RV-COV-NONE-TXT
11233         ELSE
11234             MOVE "SEE FORM"
11235             TO WS-PRT-RV-COV-NONE-TXT
11236         END-IF
11237     ELSE
11238         IF WS-TBL-LIAB-LT(WS-ENTRY-COUNT) NUMERIC
11239             MOVE WS-TBL-LIAB-LT(WS-ENTRY-COUNT)
11240             TO WS-PRT-RV-COV-LT
11241         ELSE
11242             MOVE WS-TBL-LIAB-LT(WS-ENTRY-COUNT)
11243             TO WS-PRT-RV-COV-LT-CHAR
11244         END-IF
11245     END-IF
11246 MOVE WS-LIAB-COST TO WS-PRT-RV-COV-PX
11247 MOVE WS-LIAB-COST-NET TO WS-PRT-RV-COV-PX-NET
11248 ELSE
11249     MOVE "NOT APPLICABLE"
11250     TO WS-PRT-RV-COV-NONE-TXT

```

```

11251      END-IF.
11252      WRITE PRINT-LINE  FROM WS-PRT-RV-COV
11253          AFTER ADVANCING 2 LINES.
11254      *
11255      * PRINT PIP LINE
11256      *
11257      MOVE SPACES          TO PRINT-LINE.
11258      MOVE SPACES          TO WS-PRT-RV-COV.
11259      MOVE "PERSONAL INJURY PROTECTION" TO WS-PRT-RV-COV-DESC.
11260      WRITE PRINT-LINE  FROM WS-PRT-RV-COV
11261          AFTER ADVANCING 2 LINES.
11262      *
11263      MOVE SPACES          TO PRINT-LINE.
11264      MOVE SPACES          TO WS-PRT-RV-COV.
11265      MOVE " NO FAULT COVERAGE" TO WS-PRT-RV-COV-DESC.
11266      MOVE "$"             TO WS-PRT-RV-COV-DOLLAR.
11267      MOVE "$"             TO WS-PRT-RV-COV-DOLLAR-1.
11268
11269      IF WS-PIP-EXIST(WS-ENTRY-COUNT) = 'Y'
11270          IF WS-GARAGE-STATE = "HI" THEN
11271              MOVE "SEE ENDORSEMENT" TO WS-PRT-RV-COV-NONE-TXT
11272          ELSE
11273              MOVE "NO DEDUCTIBLE" TO WS-PRT-RV-COV-NONE-TXT
11274          END-IF
11275          MOVE WS-TBL-PIP-WRIT(WS-ENTRY-COUNT)
11276              TO WS-PRT-RV-COV-PX
11277          MOVE WS-TBL-PIP-NET(WS-ENTRY-COUNT)
11278              TO WS-PRT-RV-COV-PX-NET
11279      ELSE
11280          IF WS-NY-PIP-EXIST(WS-ENTRY-COUNT) = 'Y'
11281              MOVE "NO DEDUCTIBLE" TO WS-PRT-RV-COV-NONE-TXT
11282              MOVE WS-TBL-NY-PIP-WRIT(WS-ENTRY-COUNT)
11283                  TO WS-PRT-RV-COV-PX
11284              MOVE WS-TBL-PIP-NET(WS-ENTRY-COUNT)
11285                  TO WS-PRT-RV-COV-PX-NET
11286          ELSE
11287              MOVE "NOT APPLICABLE" TO WS-PRT-RV-COV-NONE-TXT
11288          END-IF
11289      END-IF.
11290
11291      WRITE PRINT-LINE  FROM WS-PRT-RV-COV
11292          AFTER ADVANCING 1 LINES.
11293      *
11294      * PRINT ADDED PIP LINE
11295      *
11296      MOVE SPACES          TO PRINT-LINE.
11297      MOVE SPACES          TO WS-PRT-RV-COV.
11298      MOVE " ADDITIONAL COVERAGE" TO WS-PRT-RV-COV-DESC.
11299      MOVE "$"             TO WS-PRT-RV-COV-DOLLAR.
11300      MOVE "$"             TO WS-PRT-RV-COV-DOLLAR-1.
11301      IF WS-ADPIP-EXIST(WS-ENTRY-COUNT) = 'Y'
11302          IF WS-GARAGE-STATE = "HI" THEN
11303              MOVE "SEE ENDORSEMENT" TO WS-PRT-RV-COV-NONE-TXT
11304          ELSE
11305              MOVE "SEE FORM " TO WS-PRT-RV-COV-NONE-TXT
11306          END-IF

```

```

11307         MOVE WS-TBL-ADDED-PIP-WRIT(WS-ENTRY-COUNT)
11308             TO WS-PRT-RV-COV-PX
11309         MOVE WS-TBL-ADDED-PIP-NET(WS-ENTRY-COUNT)
11310             TO WS-PRT-RV-COV-PX-NET
11311     ELSE
11312         MOVE "NOT APPLICABLE"    TO WS-PRT-RV-COV-NONE-TXT
11313     END-IF.
11314     WRITE PRINT-LINE    FROM WS-PRT-RV-COV
11315         AFTER ADVANCING 1 LINES.
11316 *
11317 * PRINT MED LINE
11318 *
11319     MOVE SPACES          TO PRINT-LINE.
11320     MOVE SPACES          TO WS-PRT-RV-COV.
11321     MOVE "MEDICAL PAYMENTS" TO WS-PRT-RV-COV-DESC.
11322     MOVE "$"             TO WS-PRT-RV-COV-DOLLAR.
11323     MOVE "$"             TO WS-PRT-RV-COV-DOLLAR-1.
11324     IF WS-MED-EXIST(WS-ENTRY-COUNT) = 'Y'
11325         IF WS-TBL-MED-PAY-LT(WS-ENTRY-COUNT) = "SEE END"
11326             MOVE "(SEE ENDORSEMENT)"
11327                 TO WS-PRT-RV-COV-NONE-TXT
11328         ELSE
11329             IF WS-TBL-MED-PAY-LT(WS-ENTRY-COUNT) NUMERIC
11330                 MOVE WS-TBL-MED-PAY-LT(WS-ENTRY-COUNT)
11331                     TO WS-PRT-RV-COV-LT
11332             ELSE
11333                 MOVE WS-TBL-MED-PAY-LT(WS-ENTRY-COUNT)
11334                     TO WS-PRT-RV-COV-LT-CHAR
11335             END-IF
11336         END-IF
11337         MOVE WS-TBL-MED-PAY-WRIT(WS-ENTRY-COUNT)
11338             TO WS-PRT-RV-COV-PX
11339         MOVE WS-TBL-MED-PAY-NET(WS-ENTRY-COUNT)
11340             TO WS-PRT-RV-COV-PX-NET
11341     ELSE
11342         MOVE "NOT APPLICABLE"
11343             TO WS-PRT-RV-COV-NONE-TXT
11344     END-IF.
11345
11346     WRITE PRINT-LINE    FROM WS-PRT-RV-COV
11347         AFTER ADVANCING 2 LINES.
11348 *
11349 * PRINT MEDEXP LINE
11350 *
11351     IF WS-HOLD-STATE = 'VA'
11352         AND WS-POLY-EFF-DATE-YYYYMMDD >= '20040201' THEN
11353             MOVE SPACES          TO PRINT-LINE
11354             MOVE SPACES          TO WS-PRT-RV-COV
11355             MOVE "MEDICAL EXPENSE AND" TO WS-PRT-RV-COV-DESC
11356             MOVE "$"             TO WS-PRT-RV-COV-DOLLAR
11357             MOVE "$"             TO WS-PRT-RV-COV-DOLLAR-1
11358             IF WS-MEDEXP-EXIST(WS-ENTRY-COUNT) = 'Y'
11359                 IF WS-TBL-MEDEXP-LT(WS-ENTRY-COUNT) NUMERIC
11360                     MOVE WS-TBL-MEDEXP-LT(WS-ENTRY-COUNT)
11361                         TO WS-PRT-RV-COV-LT
11362                 ELSE

```



```

11363         MOVE WS-TBL-MEDEXP-LT(WS-ENTRY-COUNT)
11364         TO WS-PRT-RV-COV-LT-CHAR
11365     END-IF
11366     MOVE WS-TBL-MEDEXP-WRIT(WS-ENTRY-COUNT)
11367     TO WS-PRT-RV-COV-PX
11368     MOVE WS-TBL-MEDEXP-NET(WS-ENTRY-COUNT)
11369     TO WS-PRT-RV-COV-PX-NET
11370 ELSE
11371     MOVE "NOT APPLICABLE"
11372     TO WS-PRT-RV-COV-NONE-TXT
11373 END-IF
11374
11375 WRITE PRINT-LINE FROM WS-PRT-RV-COV
11376     AFTER ADVANCING 1 LINES
11377
11378     MOVE SPACES TO PRINT-LINE
11379     MOVE SPACES TO WS-PRT-RV-COV
11380     MOVE " INCOME LOSS BENEFITS (VA ONLY)"
11381     TO WS-PRT-RV-COV-DESC
11382     WRITE PRINT-LINE FROM WS-PRT-RV-COV
11383     AFTER ADVANCING 1 LINES
11384 END-IF.
11385 *
11386 * PRINT UM LINE
11387 *
11388     MOVE SPACES TO PRINT-LINE.
11389     MOVE SPACES TO WS-PRT-RV-COV.
11390     MOVE "UNINSURED MOTORISTS" TO WS-PRT-RV-COV-DESC.
11391     MOVE "$" TO WS-PRT-RV-COV-DOLLAR.
11392     MOVE "$" TO WS-PRT-RV-COV-DOLLAR-1.
11393     IF WS-UM-EXIST(WS-ENTRY-COUNT) = 'Y'
11394         IF WS-AUCV-AU-LIMIT-SPLIT-UM = 'Y'
11395             MOVE WS-UM-LIMIT-HOLD TO WS-AUCV-AU-SPLIT-LIMIT-2
11396             PERFORM 7685-FORMAT-SPLIT-LIMIT-2 THROUGH 7685-EXIT
11397             MOVE WS-AUCV-AU-SPLIT-LIMIT-2
11398             TO WS-PRT-RV-COV-LIMIT-LNG
11399         ELSE
11400             IF WS-TBL-UM-LT(WS-ENTRY-COUNT) NUMERIC
11401                 MOVE WS-TBL-UM-LT(WS-ENTRY-COUNT)
11402                 TO WS-PRT-RV-COV-LT
11403             ELSE
11404                 MOVE WS-TBL-UM-LT(WS-ENTRY-COUNT)
11405                 TO WS-PRT-RV-COV-LT-CHAR
11406             END-IF
11407         END-IF
11408         MOVE WS-TBL-UM-NET(WS-ENTRY-COUNT)
11409         TO WS-PRT-RV-COV-PX-NET
11410         MOVE WS-TBL-UM-WRIT(WS-ENTRY-COUNT)
11411         TO WS-PRT-RV-COV-PX
11412     ELSE
11413         IF WS-SUM-EXIST(WS-ENTRY-COUNT) = 'Y'
11414             IF WS-AUCV-AU-LIMIT-SPLIT-SUM = 'Y'
11415                 MOVE WS-SUM-LIMIT-HOLD TO WS-AUCV-AU-SPLIT-LIMIT-2
11416                 PERFORM 7685-FORMAT-SPLIT-LIMIT-2 THROUGH 7685-EXIT
11417                 MOVE WS-AUCV-AU-SPLIT-LIMIT-2
11418                 TO WS-PRT-RV-COV-LIMIT-LNG

```

```

11419         ELSE
11420             IF WS-TBL-SUM-LT(WS-ENTRY-COUNT) NUMERIC
11421                 MOVE WS-TBL-SUM-LT(WS-ENTRY-COUNT)
11422                 TO WS-PRT-RV-COV-LT
11423             ELSE
11424                 MOVE WS-TBL-SUM-LT(WS-ENTRY-COUNT)
11425                 TO WS-PRT-RV-COV-LT-CHAR
11426             END-IF
11427         END-IF
11428         MOVE WS-TBL-SUM-NET(WS-ENTRY-COUNT)
11429         TO WS-PRT-RV-COV-PX-NET
11430         MOVE WS-TBL-SUM-WRIT(WS-ENTRY-COUNT)
11431         TO WS-PRT-RV-COV-PX
11432     ELSE
11433         MOVE "NOT APPLICABLE" TO WS-PRT-RV-COV-NONE-TXT
11434     END-IF
11435 END-IF.
11436 WRITE PRINT-LINE FROM WS-PRT-RV-COV
11437     AFTER ADVANCING 2 LINES.
11438 *
11439 * PRINT UI LINE
11440 *
11441     MOVE SPACES TO PRINT-LINE.
11442     MOVE SPACES TO WS-PRT-RV-COV.
11443     MOVE "UNDERINSURED MOTORISTS" TO WS-PRT-RV-COV-DESC.
11444     MOVE "$" TO WS-PRT-RV-COV-DOLLAR.
11445     MOVE "$" TO WS-PRT-RV-COV-DOLLAR-1.
11446     IF WS-UI-EXIST(WS-ENTRY-COUNT) = 'Y'
11447         IF WS-AUCV-AU-LIMIT-SPLIT-UI = 'Y'
11448             MOVE WS-UI-LIMIT-HOLD TO WS-AUCV-AU-SPLIT-LIMIT-2
11449             PERFORM 7685-FORMAT-SPLIT-LIMIT-2 THROUGH 7685-EXIT
11450             MOVE WS-AUCV-AU-SPLIT-LIMIT-2
11451             TO WS-PRT-RV-COV-LIMIT-LNG
11452         ELSE
11453             IF WS-TBL-UI-LT(WS-ENTRY-COUNT) NUMERIC
11454                 MOVE WS-TBL-UI-LT(WS-ENTRY-COUNT)
11455                 TO WS-PRT-RV-COV-LT
11456             ELSE
11457                 MOVE WS-TBL-UI-LT(WS-ENTRY-COUNT)
11458                 TO WS-PRT-RV-COV-LT-CHAR
11459             END-IF
11460         END-IF
11461         MOVE WS-TBL-UI-NET(WS-ENTRY-COUNT)
11462         TO WS-PRT-RV-COV-PX-NET
11463         MOVE WS-TBL-UI-WRIT(WS-ENTRY-COUNT)
11464         TO WS-PRT-RV-COV-PX
11465     ELSE
11466         MOVE "NOT APPLICABLE" TO WS-PRT-RV-COV-NONE-TXT
11467     END-IF.
11468     WRITE PRINT-LINE FROM WS-PRT-RV-COV
11469     AFTER ADVANCING 2 LINES.
11470 *
11471 * PRINT COMP LINE
11472 *
11473     MOVE SPACES TO PRINT-LINE.
11474     MOVE SPACES TO WS-PRT-RV-COV.

```

```

11475
11476 IF WS-GLASS-EXIST-M = 'Y'
11477     EVALUATE WS-ZIPC-STAT-GB-STATE
11478         WHEN 'AZ'
11479             MOVE "COMPREHENSIVE (INCL. GLASS)" TO WS-PRT-RV-COV-DESC
11480         WHEN 'CT'
11481             MOVE "COMPREHENSIVE (INCL. GLASS)" TO WS-PRT-RV-COV-DESC
11482         WHEN 'MN'
11483             MOVE "COMPREHENSIVE (INCL. GLASS)" TO WS-PRT-RV-COV-DESC
11484         WHEN 'NY'
11485             MOVE "COMPREHENSIVE (INCL. GLASS)" TO WS-PRT-RV-COV-DESC
11486     END-EVALUATE
11487 ELSE
11488     MOVE "COMPREHENSIVE" TO WS-PRT-RV-COV-DESC
11489 END-IF.
11490
11491 MOVE "$"          TO WS-PRT-RV-COV-DOLLAR.
11492 MOVE "$"          TO WS-PRT-RV-COV-DOLLAR-1.
11493 IF WS-COMP-EXIST(WSENTRY-COUNT) = 'Y'
11494     MOVE WS-TBL-COMP-DEDUCT(WSENTRY-COUNT)
11495         TO WS-PRT-RV-COV-DED-AMT
11496     MOVE "DEDUCTIBLE"
11497         TO WS-PRT-RV-COV-DED-TXT
11498     MOVE WS-TBL-COMP-NET(WSENTRY-COUNT)
11499         TO WS-PRT-RV-COV-PX-NET
11500     MOVE WS-TBL-COMP-WRIT(WSENTRY-COUNT)
11501         TO WS-PRT-RV-COV-PX
11502 ELSE
11503     MOVE "NOT APPLICABLE" TO WS-PRT-RV-COV-NONE-TXT
11504 END-IF.
11505 WRITE PRINT-LINE FROM WS-PRT-RV-COV
11506     AFTER ADVANCING 2 LINES.
11507 *
11508 * PRINT COLL LINE
11509 *
11510 MOVE SPACES      TO PRINT-LINE.
11511 MOVE SPACES      TO WS-PRT-RV-COV.
11512
11513 IF WS-INFC-GB-PRIMARY-STATE = 'CA'
11514     and WS-WAIVE-EXIST(WSENTRY-COUNT) = 'Y'
11515 MOVE "COLLISION (INCLUDES WAIVE)" TO WS-PRT-RV-COV-DESC
11516 ELSE IF WS-GLASS-EXIST-L = 'Y'
11517     EVALUATE WS-ZIPC-STAT-GB-STATE
11518         WHEN 'AZ'
11519             MOVE "COLLISION (INCLUDES GLASS)" TO WS-PRT-RV-COV-DESC
11520         WHEN 'CT'
11521             MOVE "COLLISION (INCLUDES GLASS)" TO WS-PRT-RV-COV-DESC
11522         WHEN 'MN'
11523             MOVE "COLLISION (INCLUDES GLASS)" TO WS-PRT-RV-COV-DESC
11524         WHEN 'NY'
11525             MOVE "COLLISION (INCLUDES GLASS)" TO WS-PRT-RV-COV-DESC
11526     END-EVALUATE
11527 ELSE
11528 MOVE "COLLISION" TO WS-PRT-RV-COV-DESC
11529 END-IF.
11530

```

```

11531      MOVE "$"          TO WS-PRT-RV-COV-DOLLAR.
11532      MOVE "$"          TO WS-PRT-RV-COV-DOLLAR-1.
11533      IF WS-COLL-EXIST(WS-ENTRY-COUNT) = 'Y'
11534          IF WS-LIMF(WS-ENTRY-COUNT) = 'Y'
11535              MOVE "NO DEDUCTIBLE" TO WS-PRT-RV-COV-NONE-TXT
11536          ELSE
11537              MOVE WS-TBL-COLL-DEDUCT(WS-ENTRY-COUNT)
11538                  TO WS-PRT-RV-COV-DED-AMT
11539              MOVE "DEDUCTIBLE"
11540                  TO WS-PRT-RV-COV-DED-TXT
11541          END-IF
11542          MOVE WS-TBL-COLL-NET(WS-ENTRY-COUNT)
11543              TO WS-PRT-RV-COV-PX-NET
11544          MOVE WS-TBL-COLL-WRIT(WS-ENTRY-COUNT)
11545              TO WS-PRT-RV-COV-PX
11546      ELSE
11547          MOVE "NOT APPLICABLE" TO WS-PRT-RV-COV-NONE-TXT
11548      END-IF.
11549      WRITE PRINT-LINE FROM WS-PRT-RV-COV
11550          AFTER ADVANCING 2 LINES.
11551  *
11552  * PRINT TOW LINE
11553  *
11554      MOVE SPACES TO PRINT-LINE.
11555      MOVE SPACES TO WS-PRT-RV-COV.
11556      MOVE "TOWING" TO WS-PRT-RV-COV-DESC.
11557      MOVE "$" TO WS-PRT-RV-COV-DOLLAR.
11558      MOVE "$" TO WS-PRT-RV-COV-DOLLAR-1.
11559      IF WS-TOW-EXIST(WS-ENTRY-COUNT) = 'Y'
11560          MOVE "$50/DISABLEMENT"
11561              TO WS-PRT-RV-COV-NONE-TXT
11562          MOVE WS-TBL-TOW-NET(WS-ENTRY-COUNT)
11563              TO WS-PRT-RV-COV-PX-NET
11564          MOVE WS-TBL-TOW-WRIT(WS-ENTRY-COUNT)
11565              TO WS-PRT-RV-COV-PX
11566      ELSE
11567          MOVE "NOT APPLICABLE" TO WS-PRT-RV-COV-NONE-TXT
11568      END-IF.
11569      WRITE PRINT-LINE FROM WS-PRT-RV-COV
11570          AFTER ADVANCING 2 LINES.
11571
11572
11573  *
11574  * RENTAL REIMBURSEMENT
11575  *
11576
11577      if WS-RENTL-EXIST(WS-ENTRY-COUNT) = 'Y'
11578          MOVE SPACES TO PRINT-LINE
11579          MOVE SPACES TO WS-PRT-RV-COV
11580          MOVE "RENTAL REIMBURSEMENT" TO WS-PRT-RV-COV-DESC
11581          MOVE "$" TO WS-PRT-RV-COV-DOLLAR
11582          MOVE "$" TO WS-PRT-RV-COV-DOLLAR-1
11583
11584              MOVE "$25DAY/30DAYS/$750"
11585                  TO WS-RENTAL-REIM-PX
11586

```

```

11587         MOVE WS-RENT-PREM-WRIT(W$-ENTRY-COUNT)
11588         TO WS-PRT-RV-COV-PX
11589         display 'rent writ ' WS-PRT-RV-COV-PX
11590
11591         MOVE WS-RENT-PREM-NET(W$-ENTRY-COUNT)
11592         TO WS-PRT-RV-COV-PX-NET
11593         display 'rent net ' WS-PRT-RV-COV-PX-NET
11594
11595         WRITE PRINT-LINE FROM WS-PRT-RV-COV
11596         AFTER ADVANCING 2 LINES
11597     end-if.
11598
11599
11600 *
11601     MOVE SPACES TO PRINT-LINE.
11602     MOVE WS-PRT-AUTO-LINE TO PRT-RV-MISC-1-TEXT.
11603     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11604 *
11605     MOVE SPACES TO PRINT-LINE.
11606     MOVE WS-LINEPTR TO PRT-MISC-INFO.
11607     WRITE PRINT-LINE.
11608 *
11609     MOVE SPACES TO PRINT-LINE.
11610     WRITE PRINT-LINE
11611     AFTER ADVANCING 4 LINES.
11612 *
11613     MOVE WS-ENTRY-COUNT TO WS-PRT-RV-NO.
11614 *
11615     if WS-POLY-BSOR-GB-FM-ACCT = 5
11616         MOVE SPACES TO PRINT-LINE
11617         MOVE " State National Insurance Company, Inc."
11618         TO PRT-RV-MISC-1-LEFT
11619         MOVE WS-PRT-RV-TERMS TO PRT-RV-MISC-1-RIGHT
11620         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
11621         MOVE " AMA-111-SN (09/12)" TO WS-HOLD-EDITION
11622     else
11623         MOVE SPACES TO PRINT-LINE
11624         MOVE " FLORISTS' MUTUAL INSURANCE COMPANY"
11625         TO PRT-RV-MISC-1-LEFT
11626         MOVE WS-PRT-RV-TERMS TO PRT-RV-MISC-1-RIGHT
11627         WRITE PRINT-LINE AFTER ADVANCING 1 LINES
11628         MOVE " AMA-111(11/85)" TO WS-HOLD-EDITION
11629     end-if.
11630 *
11631     MOVE SPACES TO PRINT-LINE.
11632     MOVE " EDWARDSVILLE, ILLINOIS"
11633     TO PRT-RV-MISC-1-LEFT.
11634     MOVE WS-PRT-RV-AGENT TO PRT-RV-MISC-1-RIGHT.
11635     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11636 *
11637     MOVE SPACES TO PRINT-LINE.
11638     MOVE WS-HOLD-EDITION
11639     TO PRT-RV-MISC-1-LEFT.
11640     MOVE WS-PRT-RV-PAGE TO PRT-RV-MISC-1-RIGHT.
11641     WRITE PRINT-LINE AFTER ADVANCING 1 LINES.
11642 *

```

```

11643      MOVE SPACES      TO PRINT-LINE.
11644      MOVE WS-COURIER  TO PRT-MISC-INFO.
11645      WRITE PRINT-LINE AFTER ADVANCING 1 LINE.
11646  *
11647      ADD 1 TO WS-ENTRY-COUNT.
11648  *
11649      7800-EXIT.
11650      EXIT.
11651  *
11652  *
11653  *-----*
11654  *  THIS PARAGRAPH WILL BE USED TO SELECT THE MACRO NUMBER FOR THE *
11655  *  PAGE TO PRINTED.                                           *
11656  *-----*
11657  *
11658      8500-GET-MACRO.
11659  *
11660      INITIALIZE
11661          WS-PLFD-GB-MACRO
11662          WS-PLFD-GB-OPTION-OUT
11663          WS-PLFD-GB-FORM-NO.
11664  *
11665      EXEC SQL AT ORA1
11666          SELECT PLFD_GB_MACRO,
11667                 NVL(PLFD_GB_OPTION,'N'),
11668                 PLFD_GB_FORM_NO
11669          INTO :WS-PLFD-GB-MACRO,
11670              :WS-PLFD-GB-OPTION-OUT,
11671              :WS-PLFD-GB-FORM-NO
11672          FROM (SELECT PLFD_GB_MACRO,
11673                     PLFD_GB_OPTION,
11674                     PLFD_GB_FORM_NO
11675                FROM PLFD
11676                WHERE PLFD_GB_STATE = :WS-PLFD-GB-STATE
11677                AND PLFD_GB_FORM_DESC = :WS-PLFD-GB-FORM-DESC
11678                AND PLFD_GB_EFF_DATE
11679                <= TO_DATE(:WS-POLY-GB-EFF-DATE,'MM/DD/YYYY')
11680                AND NVL(PLFD_GB_ENABLE,'N')
11681                = :WS-PLFD-GB-ENABLE
11682                AND NVL(PLFD_GB_OPTION,'XX') IN
11683                (:WS-PLFD-GB-OPTION-1, :WS-PLFD-GB-OPTION-2)
11684                AND PLFD_BSOR_GB_FM_ACCT
11685                = :WS-POLY-BSOR-GB-FM-ACCT
11686                ORDER BY PLFD_GB_EFF_DATE DESC)
11687          WHERE ROWNUM = 1
11688      END-EXEC.
11689  *
11690      IF WS-PLFD-GB-FORM-DESC = 'AUCOV3'
11691          DISPLAY 'STATE: ' WS-PLFD-GB-STATE
11692          DISPLAY 'FORM: ' WS-PLFD-GB-FORM-DESC
11693          DISPLAY 'EFF DATE ' WS-POLY-GB-EFF-DATE
11694          DISPLAY 'ENABLED: ' WS-PLFD-GB-ENABLE
11695          DISPLAY 'FMIC CO: ' WS-POLY-BSOR-GB-FM-ACCT
11696      END-IF.
11697  *
11698  *

```



```

11699      IF    SQLCODE NOT EQUAL 0
11700      AND SQLCODE NOT EQUAL 1403
11701      MOVE "ERROR SELECTING MACRO 800 " TO WS-ERR-MSG
11702      PERFORM 9999-DISPLAY-ERROR THRU 1000-EXIT
11703      END-IF.
11704      *
11705      8500-EXIT.
11706      EXIT.
11707      *
11708      *-----*
11709      *  THIS PARAGRAPH WILL CHANGED THE PRINTER BACK TO REGULAR PRINT  *
11710      *  AND CLOSE THE PRINT FILE.                                     *
11711      *-----*
11712      *
11713      9000-CLOSE.
11714
11715      *
11716      CLOSE REPORT-OUT.
11717      *
11718      9000-EXIT.
11719      EXIT.
11720      *
11721      9500-PRT-ERR-MSG.
11722      *
11723      MOVE SPACES TO PRINT-LINE.
11724      MOVE WS-REG-FORM-ESC
11725      TO PRT-MISC-INFO.
11726      WRITE PRINT-LINE
11727      AFTER ADVANCING PAGE.
11728      *
11729      MOVE 1 TO WS-ENTRY-COUNT-6.
11730      IF WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) > 99
11731      DISPLAY "WS-ERR-MSG-TBL OCCURS 100 TIMES"
11732      ELSE
11733      PERFORM 9550-PRT-ERR-MSG-DETAIL THRU 9550-EXIT
11734      UNTIL WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) = SPACES
11735      END-IF.
11736      *
11737      9500-EXIT.
11738      EXIT.
11739      *
11740      9550-PRT-ERR-MSG-DETAIL.
11741      *
11742      IF WS-ENTRY-COUNT-6 = 1
11743      MOVE SPACES TO PRINT-LINE
11744      MOVE "WARNING!! WARNING!! HAD THE FOLLOWING ERRORS : "
11745      TO PRT-MISC-INFO
11746      WRITE PRINT-LINE AFTER ADVANCING 1 LINE
11747      END-IF.
11748      *
11749      MOVE SPACES TO PRINT-LINE.
11750      MOVE WS-ERR-MSG-TBL(WS-ENTRY-COUNT-6) TO PRT-MISC-INFO.
11751      WRITE PRINT-LINE.
11752      *
11753      ADD 1 TO WS-ENTRY-COUNT-6.
11754      *

```

```

11755 9550-EXIT.
11756 EXIT.
11757 *
11758 * -----*
11759 * THIS PARAGRAPH WILL WRITE THE ERRORS ENCOUNTERED TO THE PRINT *
11760 * THE PRINT FILE. *
11761 * -----*
11762 *
11763 9999-DISPLAY-ERROR.
11764 *
11765 MOVE SPACES TO PRINT-LINE.
11766 WRITE PRINT-LINE
11767 AFTER ADVANCING PAGE.
11768 *
11769 MOVE SPACES TO PRINT-LINE.
11770 MOVE "PLEASE FORWARD THE FOLLOWING INFORMATION TO D.P."
11771 TO PRT-MISC-INFO.
11772 WRITE PRINT-LINE
11773 AFTER ADVANCING 1 LINE.
11774 *
11775 MOVE SPACES TO PRINT-LINE.
11776 MOVE "AUTO RATING PRINT PROGRAM HAD AN ORACLE ERROR!!!"
11777 TO PRT-MISC-INFO.
11778 WRITE PRINT-LINE
11779 AFTER ADVANCING 1 LINE.
11780 *
11781 MOVE SPACES TO PRINT-LINE.
11782 MOVE "AUDECPRT HAD AN ORACLE ERROR!!!"
11783 TO PRT-MISC-INFO.
11784 WRITE PRINT-LINE
11785 AFTER ADVANCING 1 LINE.
11786 *
11787 MOVE SPACES TO PRINT-LINE.
11788 MOVE WS-ERR-MSG TO PRT-MISC-INFO.
11789 WRITE PRINT-LINE
11790 AFTER ADVANCING 1 LINE.
11791 *
11792 MOVE SPACES TO PRINT-LINE.
11793 MOVE WS-PPRT-INFC-GB-NO TO WS-DISP-NO.
11794 STRING "INFORCE_GB_NO = ",
11795 WS-DISP-NO
11796 DELIMITED BY SIZE
11797 INTO PRT-MISC-INFO.
11798 WRITE PRINT-LINE
11799 AFTER ADVANCING 1 LINE.
11800 *
11801 MOVE SPACES TO PRINT-LINE.
11802 MOVE SQLCODE TO WS-DISP-NO.
11803 STRING "SQLCODE = ",
11804 WS-DISP-NO
11805 DELIMITED BY SIZE
11806 INTO PRT-MISC-INFO.
11807 WRITE PRINT-LINE
11808 AFTER ADVANCING 1 LINE.
11809 *
11810 PERFORM 9500-PRT-ERR-MSG THRU 9500-EXIT.

```

```

11811      *
11812      PERFORM 9000-CLOSE      THRU  9000-EXIT.
11813      *      DISPLAY "ABENDSYM:YES".
11814      STOP RUN.
11815      *
11816      9999-EXIT.
11817      EXIT.
11818      *
11819      *-----*
11820      *          ***      END OF PROGRAM      ***          *
11821      *-----*
11822
11823

```