

```

1 CREATE OR REPLACE PACKAGE BODY ADD_ORS.ADD_UE AS
2 --
3 FUNCTION show_version
4     RETURN cmxlb.cmx_med_str
5 AS
6 addue_version cmxlb.cmx_med_str := 'Version 1.4 (2014-03-13 16:02:00)';
7 BEGIN
8     RETURN addue_version;
9 END;
10 --
11 FUNCTION GET_PARAMETER(in_name VARCHAR2, in_dft VARCHAR2) RETURN VARCHAR2 AS
12     result VARCHAR2(255);
13 BEGIN
14     SELECT VALUE INTO result FROM C_PARAMETERS WHERE NAME = in_name;
15     return result;
16 EXCEPTION
17     WHEN OTHERS THEN
18         RETURN in_dft;
19 END;
20 --
21 PROCEDURE chk_acct_ids
22 (
23     acct_rowid_object      varchar2,
24     acct_sys_name          varchar2,
25     acct_sys_id            varchar2,
26     acct_cd                varchar2,
27     src_nf_ok              boolean,
28     out_rowid_object OUT   varchar2,
29     out_pkey_src_obj OUT   varchar2,
30     out_err_flg           OUT   char,
31     out_err_msg           OUT   varchar2
32 )
33 AS
34 sql_stmt varchar2(2000);
35 t_src_sys_name  varchar2(10);
36 t_src_sys_id    varchar2(40);
37 t_rowid_object  varchar2(14);
38 t_count         integer;
39 BEGIN
40     IF acct_rowid_object is not null THEN
41         BEGIN
42             -- Check if the rowid_object is valid
43             sql_stmt := 'select src_sys_name, src_sys_id from c_hm_accounts where rowid_object =
''' || acct_rowid_object || ''';
44             --dbms_output.put_line('sql_stmt = ' || sql_stmt);
45             EXECUTE IMMEDIATE sql_stmt into t_src_sys_name, t_src_sys_id;
46             --dbms_output.put_line('account found');
47             -- Rowid found
48             out_rowid_object := acct_rowid_object;
49             -- Version 1.4 - Always use account rowid_object to build pkey source object
50             out_pkey_src_obj := 'BATCH' || ':' || acct_rowid_object;
51             out_err_flg := 'N';
52             out_err_msg := '';
53         EXCEPTION
54             WHEN NO_DATA_FOUND THEN
55                 --dbms_output.put_line('account not found');
56                 out_rowid_object := NULL;
57                 out_pkey_src_obj := NULL;
58                 out_err_flg := 'Y';
59                 out_err_msg := 'Invalid account rowid object: ' || acct_rowid_object;
60         END;
61     ELSIF (acct_sys_name is NOT NULL and acct_sys_id is NOT NULL) THEN

```

```

62 BEGIN
63     -- Check if there is an account with the provided source keys
64     sql_stmt := 'select rowid_object from c_hm_accounts where src_sys_name = :1 and
src_sys_id = :2';
65     dbms_output.put_line('Executing:' || sql_stmt);
66     dbms_output.put_line('Using src_sys_name = ' || acct_sys_name || ' src_sys Id: ' ||
acct_sys_id);
67     EXECUTE IMMEDIATE sql_stmt into t_rowid_object using acct_sys_name, acct_sys_id;
68     -- Rowid found
69     out_rowid_object := t_rowid_object;
70     out_pkey_src_obj := acct_sys_name || ':' || acct_sys_id;
71     out_err_flg := 'N';
72     out_err_msg := '';
73 EXCEPTION
74     WHEN NO_DATA_FOUND THEN
75         IF src_nf_ok THEN
76             -- No existing record with this key combo, treat it as a new account
77             dbms_output.put_line('No data found, src_nf_ok is true');
78             out_rowid_object := NULL;
79             out_pkey_src_obj := acct_sys_name || ':' || acct_sys_id;
80             out_err_flg := 'N';
81             out_err_msg := '';
82         ELSE
83             dbms_output.put_line('No data found, src_nf_ok is false');
84             out_rowid_object := NULL;
85             out_pkey_src_obj := NULL;
86             out_err_flg := 'Y';
87             out_err_msg := 'Account source key not found: ' || acct_sys_id || ':' ||
acct_rowid_object;
88             END IF;
89     END;
90     ELSIF acct_cd IS NOT NULL THEN
91     BEGIN
92         -- Check if there are multiple accounts with this account code
93         sql_stmt := 'select count(*) from c_hm_accounts where account_cd = :a1';
94         EXECUTE IMMEDIATE sql_stmt into t_count using acct_cd;
95         --
96         -- Bugfix for 3.11. Moved the single row select statment down so it is executed
*after* checking that the record count=1
97         --
98         IF t_count = 0 THEN
99             IF src_nf_ok THEN
100                 -- No existing record with this account code, treat it as a new account
101                 out_rowid_object := NULL;
102                 out_pkey_src_obj := 'BATCH' || ':' || acct_cd;
103                 out_err_flg := 'N';
104                 out_err_msg := '';
105             ELSE
106                 out_rowid_object := NULL;
107                 out_pkey_src_obj := NULL;
108                 out_err_flg := 'Y';
109                 out_err_msg := 'No account found for account code: ' || acct_cd;
110             END IF;
111         ELSIF t_count = 1 THEN
112             -- Get the id of the account with the provided account cd
113             sql_stmt := 'select rowid_object from c_hm_accounts where account_cd = :a1';
114             EXECUTE IMMEDIATE sql_stmt into t_rowid_object using acct_cd;
115             out_rowid_object := t_rowid_object;
116             out_pkey_src_obj := 'BATCH' || ':' || acct_cd;
117             out_err_flg := 'N';
118             out_err_msg := '';
119         ELSE

```

```

120         out_rowid_object := NULL;
121         out_pkey_src_obj := NULL;
122         out_err_flg := 'Y';
123         out_err_msg := 'More than one row found for account Code: ' || acct_cd;
124     END IF;
125     EXCEPTION
126     WHEN NO_DATA_FOUND THEN
127         IF src_nf_ok THEN
128             -- No existing record with this account code, treat it as a new account
129             out_rowid_object := NULL;
130             out_pkey_src_obj := 'BATCH' || ':' || acct_cd;
131             out_err_flg := 'N';
132             out_err_msg := '';
133         ELSE
134             out_rowid_object := NULL;
135             out_pkey_src_obj := NULL;
136             out_err_flg := 'Y';
137             out_err_msg := 'No account found for account code: ' || acct_cd;
138         END IF;
139     END;
140     ELSE
141         -- No account identifiers provided
142         out_rowid_object := NULL;
143         out_pkey_src_obj := NULL;
144         out_err_flg := 'Y';
145         out_err_msg := 'At least one account identifier should be provided';
146     END IF;
147     --
148     END chk_acct_ids;
149     --
150     --
151     PROCEDURE chk_ptac_ids
152     (
153         ptac_rowid_object      varchar2,
154         ptac_sys_name          varchar2,
155         ptac_sys_id            varchar2,
156         out_rowid_object OUT  varchar2,
157         out_err_flg          OUT  char,
158         out_err_msg          OUT  varchar2
159     )
160     AS
161     sql_stmt varchar2(2000);
162     t_src_sys_name  varchar2(10);
163     t_src_sys_id    varchar2(40);
164     t_rowid_object  varchar2(14);
165     t_count         integer;
166     BEGIN
167         IF ptac_rowid_object is not null THEN
168             BEGIN
169                 -- Check if the rowid_object is valid
170                 sql_stmt := 'select rowid_object from c_bo_party_acct where rowid_object = :rid';
171                 EXECUTE IMMEDIATE sql_stmt into t_rowid_object using ptac_rowid_object;
172                 -- Rowid found
173                 out_rowid_object := ptac_rowid_object;
174                 out_err_flg := 'N';
175                 out_err_msg := '';
176             EXCEPTION
177             WHEN NO_DATA_FOUND THEN
178                 out_rowid_object := NULL;
179                 out_err_flg := 'Y';
180                 out_err_msg := 'Invalid primary customer rowid object: ' || ptac_rowid_object;
181         END;

```

```

182     ELSIF (ptac_sys_name is NOT NULL and ptac_sys_id is NOT NULL) THEN
183     BEGIN
184         -- Check if there is an account with the provided source keys
185         sql_stmt := 'select rowid_object from c_bo_party_acct_xref where rowid_system = ''' ||
ptac_sys_name ||
186                     ''' and pkey_src_object = ''' || ptac_sys_id || '''';
187         --dbms_output.put_line('Executing:' || sql_stmt);
188         --dbms_output.put_line('Using rowid_system = ' || ptac_sys_name || ' Sys Id: ' ||
ptac_sys_id);
189         --EXECUTE IMMEDIATE sql_stmt into t_rowid_object using ptac_sys_name, ptac_sys_id;
190         EXECUTE IMMEDIATE sql_stmt into t_rowid_object;
191         -- Rowid found
192         out_rowid_object := t_rowid_object;
193         out_err_flg := 'N';
194         out_err_msg := '';
195     EXCEPTION
196         WHEN NO_DATA_FOUND THEN
197             -- No existing PTAC record with this key combo, error
198             out_rowid_object := NULL;
199             out_err_flg := 'Y';
200             out_err_msg := 'No customer record found for provided source keys: ' ||
ptac_sys_name || ':' || ptac_sys_id;
201     END;
202     ELSE
203         -- No PTAC identifiers provided
204         out_rowid_object := NULL;
205         out_err_flg := 'N';
206         out_err_msg := '';
207     END IF;
208 END chk_ptac_ids;
209
210
211 PROCEDURE post_landing
212 (
213     in_rowid_job          cmxlb.cmx_rowid,
214     in_ldg_table_name     cmxlb.cmx_table_name,
215     in_stg_table_name     cmxlb.cmx_table_name,
216     in_prl_table_name     cmxlb.cmx_table_name,
217     out_error_msg         OUT cmxlb.cmx_message,
218     out_return_code       OUT int
219 )
220 AS
221 t_acc_rowid_object varchar2(14);
222 t_acc_rowid_object2 varchar2(14);
223 t_ptac_rowid_object varchar2(14);
224 t_pkey_src_obj varchar2(100);
225 t_pkey_src_obj2 varchar2(100);
226 t_err_flg char(1);
227 t_err_msg varchar2(2000);
228 --
229 TYPE ref_cursor is REF CURSOR;
230 m_cursor ref_cursor;
231 --
232 cursor c_accts is
233 select acc_rowid_object, account_cd, src_sys_name, src_sys_id, ptac_rowid_object,
ptac_src_name, ptac_src_key
234 from c_ldg_hm_accounts
235 for update;
236 --
237 cursor c_accts_rel is
238 select acc_rowid_object_1, account_cd_1, src_sys_name_1, src_sys_id_1,
239        acc_rowid_object_2, account_cd_2, src_sys_name_2, src_sys_id_2

```

```

240 from c_ldg_accounts_rel
241 for update;
242 --
243 cursor c_accts_ptac_rel is
244 select acc_rowid_object, account_cd, src_sys_name, src_sys_id, ptac_rowid_object,
ptac_src_name, ptac_src_key
245 from C_LDG_HM_ACCT_PTY_AC_REL
246 for update;
247 --
248 --
249 BEGIN
250     cmxlog.debug ('Inside ADD_UE.post_landing');
251     IF in_ldg_table_name = 'C_LDG_HM_ACCOUNTS' AND in_stg_table_name =
'C_STG_HM_ACCTS_BATCH' THEN
252         BEGIN
253             cmxlog.debug ('ADDUE: Landing table name is ' || in_ldg_table_name || ' Staging table
name is ' || in_stg_table_name);
254             FOR r_accts in c_accts LOOP
255                 --Check the account identifiers
256                 chk_acct_ids (r_accts.acc_rowid_object, r_accts.src_sys_name, r_accts.src_sys_id,
r_accts.account_cd, TRUE, t_acc_rowid_object, t_pkey_src_obj, t_err_flg, t_err_msg);
257                 IF t_err_flg = 'N' THEN
258                     chk_ptac_ids (r_accts.ptac_rowid_object, r_accts.ptac_src_name, r_accts.
ptac_src_key, t_ptac_rowid_object, t_err_flg, t_err_msg);
259                 END IF;
260                 UPDATE C_LDG_HM_ACCOUNTS
261                 SET     cmp_acc_rowid_object = t_acc_rowid_object,
262                        cmp_pkey_src_obj = t_pkey_src_obj,
263                        cmp_ptac_rowid_object = t_ptac_rowid_object,
264                        error_flag = t_err_flg,
265                        error_msg = t_err_msg
266                 WHERE CURRENT OF c_accts;
267             END LOOP;
268             COMMIT;
269         END;
270     ELSIF in_ldg_table_name = 'C_LDG_ACCOUNTS_REL' AND in_stg_table_name =
'C_STG_HM_ACCTS_REL_BATCH' THEN
271         BEGIN
272             cmxlog.debug ('ADDUE: Landing table name is ' || in_ldg_table_name || ' Staging table
name is ' || in_stg_table_name);
273             FOR r_accts_rel in c_accts_rel LOOP
274                 --Check the account identifiers
275                 chk_acct_ids (r_accts_rel.acc_rowid_object_1, r_accts_rel.src_sys_name_1,
r_accts_rel.src_sys_id_1, r_accts_rel.account_cd_1, FALSE, t_acc_rowid_object,
t_pkey_src_obj, t_err_flg, t_err_msg);
276                 IF t_err_flg = 'N' THEN
277                     chk_acct_ids (r_accts_rel.acc_rowid_object_2, r_accts_rel.src_sys_name_2,
r_accts_rel.src_sys_id_2, r_accts_rel.account_cd_2, FALSE, t_acc_rowid_object2,
t_pkey_src_obj2, t_err_flg, t_err_msg);
278                 END IF;
279                 UPDATE C_LDG_ACCOUNTS_REL
280                 SET     cmp_acc_rowid_object_1 = t_acc_rowid_object,
281                        cmp_pkey_src_obj = t_pkey_src_obj || ':' || t_pkey_src_obj2,
282                        cmp_acc_rowid_object_2 = t_acc_rowid_object2,
283                        error_flag = t_err_flg,
284                        error_msg = t_err_msg
285                 WHERE CURRENT OF c_accts_rel;
286             END LOOP;
287             COMMIT;
288         END;
289     ELSIF in_ldg_table_name = 'C_LDG_HM_ACCT_PTY_AC_REL' AND in_stg_table_name =
'C_STG_HM_ACCT_PTAC_BATCH' THEN

```

```

290 BEGIN
291     cmxlog.debug ('ADDUE: Landing table name is ' || in_ldg_table_name || ' Staging table
name is ' || in_stg_table_name);
292     FOR r_accts_ptac_rel in c_accts_ptac_rel LOOP
293         --Check the account identifiers
294         chk_acct_ids (r_accts_ptac_rel.acc_rowid_object, r_accts_ptac_rel.src_sys_name,
r_accts_ptac_rel.src_sys_id, r_accts_ptac_rel.account_cd, FALSE, t_acc_rowid_object,
t_pkey_src_obj, t_err_flg, t_err_msg);
295         --
296         IF t_err_flg = 'N' THEN
297             chk_ptac_ids (r_accts_ptac_rel.ptac_rowid_object, r_accts_ptac_rel.ptac_src_name
, r_accts_ptac_rel.ptac_src_key, t_ptac_rowid_object, t_err_flg, t_err_msg);
298         END IF;
299         --
300         --
301         IF r_accts_ptac_rel.ptac_src_name is NOT NULL and r_accts_ptac_rel.ptac_src_key IS
NOT NULL THEN
302             t_pkey_src_obj2 := t_pkey_src_obj || ':' || r_accts_ptac_rel.ptac_src_name ||
':' || r_accts_ptac_rel.ptac_src_key;
303         ELSE
304             t_pkey_src_obj2 := t_pkey_src_obj || ':' || t_ptac_rowid_object;
305         END IF;
306         --
307         --
308         UPDATE C_LDG_HM_ACCT_PTY_AC_REL
309         SET     cmp_acc_rowid_object = t_acc_rowid_object,
310               cmp_pkey_src_obj = t_pkey_src_obj2,
311               cmp_ptac_rowid_object = t_ptac_rowid_object,
312               error_flag = t_err_flg,
313               error_msg = t_err_msg
314         WHERE CURRENT OF c_accts_ptac_rel;
315     END LOOP;
316     COMMIT;
317 END;
318 ELSE
319     -- Do nothing
320     CMXlog.debug ('ADDUE Post Landing - no action taken');
321 END IF;
322 END;
323 PROCEDURE post_stage_concat
324 (
325     in_party_acct_id varchar2,
326     out_txn_div_display OUT varchar2
327 )
328 AS
329 --
330 BEGIN
331     with
332     stgbo as
333     (
334         select party_acct_id, TXN_DIV_CD from C_BO_PTAC_TXN_DIV WHERE party_acct_id =
in_party_acct_id
335     union
336         select party_acct_id, TXN_DIV_CD from C_STG_PTAC_TXN_DIV WHERE PARTY_ACCT_ID =
in_party_acct_id
337     )
338     select distinct TRIM(A||' '||B||' '||C||' '||D) into out_txn_div_display from (select
party_acct_id, TXN_DIV_CD from stgbo ) pivot( max(TXN_DIV_CD) for TXN_DIV_CD in ( 'ADC' A,
'ADD' B, 'AMD' C, 'APOC' D)) WHERE PARTY_ACCT_ID = in_party_acct_id;
339 END post_stage_concat;
340 PROCEDURE post_stage
341 (

```

```

342         in_rowid_job                cmxlb.cmx_rowid,
343         in_ldg_table_name            cmxlb.cmx_table_name,
344         in_stg_table_name            cmxlb.cmx_table_name,
345         out_error_msg                OUT cmxlb.cmx_message,
346         out_return_code              OUT int
347     )
348 AS
349     sql_stmt varchar2(2000);
350     t_party_acct_id varchar2(14);
351     t_txn_div_cd varchar2(20);
352     t_txn_div_display varchar2(50);
353     commit_count NUMBER := 0;
354     commit_inc NUMBER := 1000;
355     --
356     CURSOR C_PTAC_TXN IS
357     SELECT PARTY_ACCT_ID, TXN_DIV_CD, TXN_DIV_DISPLAY
358     FROM    C_STG_PTAC_TXN_DIV;
359     --
360     BEGIN
361     --
362         commit_inc := to_number(GET_PARAMETER('post_stage_commit', commit_inc));
363         IF in_ldg_table_name = 'C_LDG_PTAC_TXN_DIV' AND in_stg_table_name = 'C_STG_PTAC_TXN_DIV'
364         THEN
365             -- 20130225 SCK Update the stage txn_div_display col with a denormalized string
366             -- derived from an aggregate of both staging and base object
367             cmxlog.debug ('ADDUE: Landing table name is ' || in_ldg_table_name || ' Staging
368             table name is ' || in_stg_table_name);
369             BEGIN
370                 FOR R_PTAC_TXN in C_PTAC_TXN LOOP
371                     post_stage_concat(R_PTAC_TXN.PARTY_ACCT_ID, t_txn_div_display);
372                     UPDATE C_STG_PTAC_TXN_DIV
373                     SET txn_div_display = t_txn_div_display, create_date = sysdate WHERE
374                     TXN_DIV_CD = R_PTAC_TXN.TXN_DIV_CD AND
375                     PARTY_ACCT_ID = R_PTAC_TXN.PARTY_ACCT_ID; -- CURRENT OF C_PTAC_TXN;
376                     commit_count := commit_count + commit_inc;
377                     IF MOD(commit_count, 1000) = 0 THEN
378                         cmxlog.debug ('ADDUE: post_stage_concat is: ' || commit_count || ':'
379                         || R_PTAC_TXN.PARTY_ACCT_ID || ' : ' || t_txn_div_display);
380                         COMMIT;
381                     END IF;
382                 END LOOP;
383                 COMMIT;
384             END;
385         ELSE
386             CMXlog.debug ('ADDUE Post Stage - no action taken');
387         END IF;
388     END post_stage;
389 END ADD_UE;
390 /

```