|  |  |
| --- | --- |
| **Date:** | **Sep 15, 2011** |
| **Remedy Request ID:** | **Not Applicable** |
| **Production Build ID:** | **P.1.112.0** |
| Build Description : |  |

**Section 1 –**

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Build : | * Hot Fix | * Full |  |
| Database Team Responsible Party: | SACWIS DBA’s | | |

**Section 2 – Technical Solution**

|  |  |
| --- | --- |
| Location of Solution Documentation: | R:\sacwis\Database\DBAWorkArea\DBDevelopment\P.1.112.0 |
| Solution Documentation Folder Name: | Prod\Docs |
| Software/SQL File Name: | sacwis\_prod\_db\_p.1.112. |

|  |  |
| --- | --- |
| List new table(s)/View/MView be added: |  |
| List new column(s) be added / modified: | **Add 3 New Table(s)**  CREATE TABLE RB\_REQUEST  ( RB\_REQUEST\_ID NUMBER NOT NULL,  SUBMISSION\_MONTH INTEGER NULL,  SUBMISSION\_YEAR INTEGER NULL,  REQUEST\_STATUS\_CODE VARCHAR2 (30) NULL,  REQUEST\_STATE\_DATE DATE NULL,  SUBMITTED\_BY NUMBER NULL,  SUBMITTED\_DATE DATE NULL,  CREATED\_BY NUMBER NULL,  CREATED\_DATE DATE NULL,  MODIFIED\_BY NUMBER NULL,  MODIFIED\_DATE DATE NULL)  CREATE TABLE RB\_SUBMISSION (  RB\_SUBMISSION\_ID NUMBER NOT NULL,  RB\_REQUEST\_ID NUMBER NOT NULL,  RB\_PERSON\_ID NUMBER NOT NULL,  RB\_PROVIDER\_ID NUMBER NOT NULL,  FIRST\_NAME VARCHAR2 (35) NOT NULL,  MIDDLE\_NAME VARCHAR2 (25) NULL,  LAST\_NAME VARCHAR2 (35) NOT NULL,  DOB DATE NOT NULL,  SSN VARCHAR2 (11) NOT NULL,  DRIVERS\_LIC\_NUM VARCHAR2 (15) NULL,  DRIVERS\_LIC\_STATE VARCHAR2 (2) NULL,  SUBMISSION\_MONTH INTEGER NOT NULL,  SUBMISSION\_YEAR INTEGER NOT NULL,  RECORD\_FLAG VARCHAR2 (1) NOT NULL,  SUBMIT\_FLAG NUMBER NOT NULL,  RECORD\_REASON VARCHAR2 (50) NOT NULL,  RECORD\_GROUP VARCHAR2 (20) NOT NULL,  RESUBMIT\_FLAG NUMBER NULL,  RESUBMIT\_COMMENT VARCHAR2 (1500) NULL,  RESEARCH\_FLAG NUMBER NULL,  RESEARCH\_COMMENT VARCHAR2 (1500) NULL,  OVERRIDE\_FLAG NUMBER NULL,  OVERRIDE\_COMMENT VARCHAR2 (1500) NULL,  CREATED\_BY NUMBER NOT NULL,  CREATED\_DATE DATE NOT NULL,  MODIFIED\_BY NUMBER NOT NULL,  MODIFIED\_DATE DATE NOT NULL)  CREATE TABLE AGENCY\_CONFIG (  AGENCY\_CONFIG\_ID NUMBER NOT NULL,  AGENCY\_ID NUMBER NOT NULL,  CONFIG\_KEY VARCHAR2 (30) NOT NULL,  CONFIG\_VALUE VARCHAR2 (100) NOT NULL,  EFFECTIVE\_DATE DATE NOT NULL,  END\_DATE DATE NOT NULL,  CREATED\_BY NUMBER NOT NULL,  CREATED\_DATE DATE NOT NULL,  MODIFIED\_BY NUMBER NOT NULL,  MODIFIED\_DATE DATE NOT NULL)  **Modified 4 Existing Table(s)**  ALTER TABLE CHARACTERISTIC MODIFY (additional\_information VARCHAR2(4000))  ALTER TABLE CASE\_PLAN MODIFY (signature\_comment\_text VARCHAR2 (4000))  ALTER TABLE CASE\_REVIEW MODIFY (signature\_comment\_text VARCHAR2 (4000))  ALTER TABLE CR\_SIGNATURE\_DETAIL ADD (notified\_by\_mail\_date DATE NULL) |
| List existing table(s) to be deleted: |  |
| List existing column(s) to be deleted: | **Deleted Columns from 1 Existing Table(s)**  alter table  PAYMENT\_REQUEST  drop  (disbursement\_flag, disbursement\_name\_text, roster\_name\_text,  roster\_cat\_name\_text, roster\_nmd\_by\_wkr\_id, roster\_apr\_by\_wkr\_id, roster\_apr\_date) |

|  |  |
| --- | --- |
| Application Ref Data / Seed Data | **Some Ref & Seed Data Updates and Modifies:**  **Adds:**  **REF\_DATA\_CONFIG**  insert into ref\_data\_config (REF\_DATA\_CONFIG\_ID, CATEGORY\_NAME, DOMAIN\_CODE, PARENT\_DOMAIN\_CODE,  HAS\_CHILD\_FLAG, SQL\_STATEMENT,  CLASS\_TEXT, CATEGORY\_DESC, STATE\_EXCLUSIVE\_FLAG, CREATED\_BY, CREATED\_DATE, MODIFIED\_BY,  MODIFIED\_DATE)  values (REF\_DATA\_CONFIG\_ID\_SEQ.NEXTVAL, 'HmStdyVerifStatusCodeResident', 'HmStdyVerifStatusCodeResident', NULL,  NULL, 'Deprecated',  'us.oh.state.odjfs.sacwis.framework.rda.ReferenceVO', NULL, 0, 100, SYSDATE, 100, SYSDATE)  **REF\_DATA**  insert into ref\_data (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id, group\_code) values (ref\_data\_id\_seq.nextval, null,  'HmStdyVerifStatusCodeResident', 'Approved', 'Approved', 'Approved', 1, 1,  100, sysdate, 100, sysdate, (select ref\_data\_config\_id from ref\_data\_config where domain\_code = 'HmStdyVerifStatusCodeResident'),'00')  insert into ref\_data (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id, group\_code) values (ref\_data\_id\_seq.nextval, null,  'HmStdyVerifStatusCodeResident', 'NotApplicable', 'Not Applicable', 'Not Applicable', 2, 1,  100, sysdate, 100, sysdate, (select ref\_data\_config\_id from ref\_data\_config where domain\_code = 'HmStdyVerifStatusCodeResident'),  '00')  insert into ref\_data (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id, group\_code) values (ref\_data\_id\_seq.nextval, null,  'HmStdyVerifStatusCodeResident', 'NotApproved', 'Not Approved', 'Not Approved', 3, 1,  100, sysdate, 100, sysdate, (select ref\_data\_config\_id from ref\_data\_config where domain\_code = 'HmStdyVerifStatusCodeResident'), '00')  insert into ref\_data (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id, group\_code) values (ref\_data\_id\_seq.nextval, null,  'HmStdyVerifStatusCodeResident', 'Verified', 'Verified', 'Verified', 4, 1,  100, sysdate, 100, sysdate, (select ref\_data\_config\_id from ref\_data\_config where domain\_code = 'HmStdyVerifStatusCodeResident'),'00')  insert into ref\_data (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id, group\_code) values (ref\_data\_id\_seq.nextval, null,  'HmStdyVerifStatusCodeResident', 'PENDING', 'Pending', 'Pending', 5, 0,  100, sysdate, 100, sysdate, (select ref\_data\_config\_id from ref\_data\_config where domain\_code = 'HmStdyVerifStatusCodeResident'), '00')  insert into ref\_data (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id, group\_code) values (ref\_data\_id\_seq.nextval, null,  'HmStdyVerifStatusCodeResident', 'YesVerified', 'Yes-Verified', 'Yes-Verified', 6, 0,  100, sysdate, 100, sysdate, (select ref\_data\_config\_id from ref\_data\_config where domain\_code = 'HmStdyVerifStatusCodeResident'),'00')  insert into ref\_data (ref\_data\_id, parent\_ref\_data\_code, domain\_code, ref\_data\_code, short\_desc, long\_desc, sort\_value, inactive\_flag,  created\_by, created\_date, modified\_by, modified\_date, ref\_data\_config\_id, group\_code) values (ref\_data\_id\_seq.nextval, null,  'HmStdyVerifStatusCodeResident', 'NoVerified', 'No-Verified', 'No-Verified', 7, 0,  100, sysdate, 100, sysdate, (select ref\_data\_config\_id from ref\_data\_config where domain\_code = 'HmStdyVerifStatusCodeResident'),'00');  **MENU**  INSERT INTO MENU VALUES (MENU\_SEQ.NEXTVAL, 'rapbackList.do', 'Administration.Utilities', 31, 6)  **Changes:**  **REF\_DATA**  update ref\_data set long\_desc = long\_desc || ' (Optional - will not print in home study.)', modified\_by = 999, modified\_date = sysdate  where domain\_code = 'PrvdrFmlyFamilyNarrativeTopic' and ref\_data\_code = 'FAMILYNARRATIVETOPIC17'  **GROUP\_CATEGORY**  **BUSINESS\_FUNCTION**  **ROLE / SECURITY\_PROFILE**  **UC\_ACTION**  **ACTION\_ROLE**  **SECURITY\_PROFILE**  **PROFILE\_RESOURCE**  **REPORT\_TEMPLATE**  **GROUP\_CAT\_CEIL\_TYP\_LINK**  **TICKLER\_TEMPLATE**   1. **Deletes:** |
| Batch Ref Data / Seed Data |  |
| Materialized View |  |
| Sequence | **Added 3 New Sequence(s) Overall:**  CREATE SEQUENCE RB\_REQUEST\_ID\_SEQ  CREATE SEQUENCE RB\_SUBMISSION\_ID\_SEQ  CREATE SEQUENCE AGENCY\_CONFIG\_ID\_SEQ  **Modified 0 Sequence(s) Overall:** |

|  |  |
| --- | --- |
| Index | **Add 3 New Index(es) Overall:**  CREATE UNIQUE INDEX RB\_REQUEST\_P ON RB\_REQUEST  (RB\_REQUEST\_ID)  CREATE INDEX RB\_REQUEST\_CREATED\_BY ON RB\_REQUEST (CREATED\_BY)  CREATE UNIQUE INDEX RB\_SUBMISSION\_P ON RB\_SUBMISSION  (RB\_SUBMISSION\_ID)  CREATE INDEX RB\_SUBMISSION\_F01\_RB\_REQUEST ON RB\_SUBMISSION (RB\_REQUEST\_ID)  CREATE UNIQUE INDEX AGENCY\_CONFIG\_P ON AGENCY\_CONFIG  (AGENCY\_CONFIG\_ID)  CREATE INDEX AGENCY\_CONFIG\_F01\_AGENCY ON AGENCY\_CONFIG (AGENCY\_ID)  **Modified Index(es) Overall:**  **Dropped 3 Index(es) Overall:**  DROP INDEX PAYMENT\_REQUEST\_N12  DROP INDEX PAYMENT\_REQUEST\_N15  DROP INDEX PAYMENT\_REQUEST\_N17 |
| Constraint | **Added 8 New and Modified 2 Existing Constraint(s):**  ALTER TABLE RB\_REQUEST ADD (  CONSTRAINT RB\_REQUEST\_P  PRIMARY KEY  (RB\_REQUEST\_ID)  ALTER TABLE RB\_SUBMISSION ADD (  CONSTRAINT RB\_SUBMISSION\_P  PRIMARY KEY  (RB\_SUBMISSION\_ID)  ALTER TABLE RB\_SUBMISSION  ADD (CONSTRAINT RB\_SUBMISSION\_F01\_RB\_REQUEST  FOREIGN KEY (RB\_REQUEST\_ID)  REFERENCES RB\_REQUEST (RB\_REQUEST\_ID)  ALTER TABLE AGENCY\_CONFIG ADD (  CONSTRAINT AGENCY\_CONFIG\_P  PRIMARY KEY  (AGENCY\_CONFIG\_ID)  ALTER TABLE AGENCY\_CONFIG  ADD CONSTRAINT AGENCY\_CONFIG\_F01\_AGENCY  FOREIGN KEY (AGENCY\_ID)  REFERENCES AGENCY (AGENCY\_ID)  ALTER TABLE sacwis.justification  ADD CONSTRAINT intake\_f01\_justification  FOREIGN KEY (intake\_id)  REFERENCES sacwis.intake (intake\_id)  ALTER TABLE intake\_case\_link ADD CONSTRAINT intake\_case\_link\_pk0 CHECK ( intake\_case\_link\_id > 0)  ALTER TABLE sacwis.intake\_case\_link  ADD CONSTRAINT case\_base\_f01\_intake\_link  FOREIGN KEY (case\_id)  REFERENCES sacwis.case\_base (case\_id)  **Modified 2 Constraint(s) Overall:**  ALTER TABLE sacwis.intake\_case\_link  MODIFY case\_id CONSTRAINT intake\_case\_link\_case\_id\_nn NOT NULL  ALTER TABLE sacwis.intake\_case\_link  MODIFY link\_date CONSTRAINT intake\_case\_link\_link\_date\_nn NOT NULL  **Dropped 0 Constraint(s) Overall:** |
|  |  |
| Function Updated | **Added 0 New & Modified 1 Existing Function(s) & Dropped 0 Functions Overall:**  **New Function(s):**      **Updated 4 Existing Function(s):**   1. GET\_CHILD\_REL\_PERSON\_ID   **Dropped 0 Unused Function(s) :** |
| Package(s) Updated | **Added 0 New & Modified 7 Package(s) & Dropped 0 Packages Overall:**  **New Package(s):**  **Updated 7 Existing Package(s):**   1. PKINTAKEPARTICIPANT 2. FM25A\_MEDICAID\_HMO\_MITS 3. RPT\_4280\_4281\_PACK 4. ONLINE\_PROVIDER\_MERGE 5. PKG\_CRIS13\_RECONVERT 6. PKG\_NON\_ODJFS\_PRVDR\_MERGE 7. DSBRS\_WARRANT\_TO\_CAS   **Dropped 0 Unused Package(s) :** |
| Procedure to be compiled | **Modified 1 Existing Procedure(s) & Dropped 0 Existing Overall:**  **Updated Existing Procedure:**   1. SP\_MP\_UPDATEPERSONRELATEDROWS   **Dropped 0 Unused Procedure(s) :** |

|  |  |
| --- | --- |
| Triggers to be updated and enabled | **Added 3 New and Modified 2 Existing Trigger(s) Overall:Added New Trigger(s):**   1. AGENCY\_CONFIG\_AUDR 2. RB\_REQUEST\_AUDR 3. RB\_SUBMISSION\_AUDR   **Modified Existing Trigger(s):**   1. PAYMENT\_REQUEST\_AUDR 2. CR\_SIGNATURE\_DETAIL\_AUDR |
| Script Updated | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DEFECT Case Plan Signature Combination - This fix:  1) combine each case plan's signature record's non-involvement\_comment\_text  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  SET LINESIZE 200  SET PAGESIZE 1;  SET HEADING OFF;  SET FEEDBACK OFF  SET PAGES 1;  SPOOL cp\_signature\_combination\_fix.LOG;  SET serveroutput ON SIZE 1000000;  --------------------------------------------------------------  -- Drop Temporary Tables  --------------------------------------------------------------  --SET termout OFF  --DROP TABLE tmp\_cp\_sig\_comments;  --SET termout ON  --------------------------------------------------------------  -- Create Temporary Tables  --------------------------------------------------------------    --CREATE TABLE tmp\_cp\_sig\_comments (  -- case\_plan\_id NUMBER NULL,  -- signature\_count NUMBER NULL,  -- combined\_comments VARCHAR2(4000) NULL  --);  -------------------------------------------------------------------------------  DECLARE  sys\_scriptprefix VARCHAR2(35) := '[CP\_SIGNATURE\_COMBINATION\_FIX]: ';  sys\_scriptsuccess NUMBER := 1;  sys\_rowupdated NUMBER := 0;  sys\_rowdeleted NUMBER := 0;  sys\_rowinserted NUMBER := 0;  v\_error\_cnt NUMBER := 0;  v\_case\_plan\_id NUMBER := 0;  v\_signature\_cnt NUMBER := 0;  v\_comments VARCHAR(4000) := '';  -------------------------------------------------------------------------------  CURSOR cur\_cp\_signatures IS  SELECT sd.case\_plan\_id, sd.non\_involvement\_reason\_text  FROM cp\_signature\_detail sd  WHERE sd.NON\_INVOLVEMENT\_REASON\_TEXT IS NOT NULL  ORDER BY sd.case\_plan\_id, sd.cp\_signature\_detail\_id ;  -------------------------------------------------------------------------------  BEGIN  FOR cur\_rec IN cur\_cp\_signatures LOOP  BEGIN  IF (cur\_rec.case\_plan\_id != v\_case\_plan\_id) THEN  IF (v\_case\_plan\_id != 0) THEN  --INSERT INTO tmp\_cp\_sig\_comments VALUES (v\_case\_plan\_id, v\_signature\_cnt, v\_comments);  UPDATE case\_plan SET signature\_comment\_text = v\_comments WHERE case\_plan\_id = v\_case\_plan\_id;  sys\_rowupdated := sys\_rowupdated + 1;  END IF;    v\_case\_plan\_id := cur\_rec.case\_plan\_id;  v\_signature\_cnt := 0;  v\_comments := '';  END IF;  v\_comments := v\_comments || cur\_rec.non\_involvement\_reason\_text;  v\_signature\_cnt := v\_signature\_cnt + 1;  EXCEPTION  WHEN OTHERS THEN  --sys\_scriptsuccess := 0;  DBMS\_OUTPUT.PUT\_LINE('ERROR ON CASE\_PLAN\_ID ' || v\_case\_plan\_id || ' WITH SIGNATURE COUNT ' || v\_signature\_cnt);  DBMS\_OUTPUT.PUT\_LINE('Error ' || SQLERRM);  --GOTO end\_process;  v\_error\_cnt := v\_error\_cnt + 1;  END;  END LOOP;  --INSERT INTO tmp\_cp\_sig\_comments VALUES (v\_case\_plan\_id, v\_signature\_cnt, v\_comments);  UPDATE case\_plan SET signature\_comment\_text = v\_comments WHERE case\_plan\_id = v\_case\_plan\_id;  sys\_rowupdated := sys\_rowupdated + 1;    -------------------------------------------------------------------------------    <<end\_process>>  IF (sys\_scriptsuccess = 1) THEN  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || ' records updated: ' || sys\_rowupdated );  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || ' error number: ' || v\_error\_cnt);  COMMIT;  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || 'All Transactions have been COMMITted.');  ELSE  ROLLBACK;  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || 'Error occured; A ROLLBACK has been performed.');  END IF;  DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\* Process Completed \*\*\*\*\*\*\*\*');  --Main excepton-handling routine  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error - ' || SQLERRM);  ROLLBACK;  RETURN;  END;  /  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DEFECT Case Review Signature Combination - This fix:  1) combine each case review's signature record's non-involvement\_comment\_text  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  SET LINESIZE 200  SET PAGESIZE 1;  SET HEADING OFF;  SET FEEDBACK OFF  SET PAGES 1;  SPOOL cr\_signature\_combination\_fix.LOG;  SET serveroutput ON SIZE 1000000;  --------------------------------------------------------------  -- Drop Temporary Tables  --------------------------------------------------------------  --SET termout OFF  --DROP TABLE tmp\_cr\_sig\_comments;  --SET termout ON  --------------------------------------------------------------  -- Create Temporary Tables  --------------------------------------------------------------    --CREATE TABLE tmp\_cr\_sig\_comments (  -- case\_review\_id NUMBER NULL,  -- signature\_count NUMBER NULL,  -- combined\_comments VARCHAR2(4000) NULL  --);  -------------------------------------------------------------------------------  DECLARE  sys\_scriptprefix VARCHAR2(35) := '[CR\_SIGNATURE\_COMBINATION\_FIX]: ';  sys\_scriptsuccess NUMBER := 1;  sys\_rowupdated NUMBER := 0;  sys\_rowdeleted NUMBER := 0;  sys\_rowinserted NUMBER := 0;  v\_error\_cnt NUMBER := 0;  v\_case\_review\_id NUMBER := 0;  v\_signature\_cnt NUMBER := 0;  v\_comments VARCHAR(4000) := '';  -------------------------------------------------------------------------------  CURSOR cur\_cr\_signatures IS  SELECT sd.case\_review\_id, sd.non\_involvement\_reason\_text  FROM cr\_signature\_detail sd  WHERE sd.NON\_INVOLVEMENT\_REASON\_TEXT IS NOT NULL  ORDER BY sd.case\_review\_id, sd.cr\_signature\_detail\_id ;  -------------------------------------------------------------------------------  BEGIN  FOR cur\_rec IN cur\_cr\_signatures LOOP  BEGIN  IF (cur\_rec.case\_review\_id != v\_case\_review\_id) THEN  IF (v\_case\_review\_id != 0) THEN  --INSERT INTO tmp\_cr\_sig\_comments VALUES (v\_case\_review\_id, v\_signature\_cnt, v\_comments);  UPDATE case\_review SET signature\_comment\_text = v\_comments WHERE case\_review\_id = v\_case\_review\_id;  sys\_rowupdated := sys\_rowupdated + 1;  END IF;    v\_case\_review\_id := cur\_rec.case\_review\_id;  v\_signature\_cnt := 0;  v\_comments := '';  END IF;  v\_comments := v\_comments || cur\_rec.non\_involvement\_reason\_text;  v\_signature\_cnt := v\_signature\_cnt + 1;  EXCEPTION  WHEN OTHERS THEN  --sys\_scriptsuccess := 0;  DBMS\_OUTPUT.PUT\_LINE('ERROR ON CASE\_REVIEW\_ID ' || v\_case\_review\_id || ' WITH SIGNATURE COUNT ' || v\_signature\_cnt);  DBMS\_OUTPUT.PUT\_LINE('Error ' || SQLERRM);  --GOTO end\_process;  v\_error\_cnt := v\_error\_cnt + 1;  END;  END LOOP;  --INSERT INTO tmp\_cr\_sig\_comments VALUES (v\_case\_review\_id, v\_signature\_cnt, v\_comments);  UPDATE case\_review SET signature\_comment\_text = v\_comments WHERE case\_review\_id = v\_case\_review\_id;  sys\_rowupdated := sys\_rowupdated + 1;    -------------------------------------------------------------------------------    <<end\_process>>  IF (sys\_scriptsuccess = 1) THEN  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || ' records updated: ' || sys\_rowupdated );  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || ' error number: ' || v\_error\_cnt);  COMMIT;  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || 'All Transactions have been COMMITted.');  ELSE  ROLLBACK;  DBMS\_OUTPUT.PUT\_LINE(sys\_scriptprefix || 'Error occured; A ROLLBACK has been performed.');  END IF;  DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\* Process Completed \*\*\*\*\*\*\*\*');  --Main excepton-handling routine  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error - ' || SQLERRM);  ROLLBACK;  RETURN;  END;  /  UPDATE intake\_case\_link icl  SET link\_date = (SELECT NVL(i.decision\_date, icl.created\_date) FROM intake i WHERE i.intake\_id = icl.intake\_id)  WHERE link\_date IS NULL  UPDATE intake\_case\_link  SET intake\_case\_link\_id = intake\_case\_link\_id\_seq.NEXTVAL  WHERE intake\_case\_link\_id = 0  CREATE TABLE datafix\_13752\_cp\_signature  AS SELECT cp\_signature\_detail\_id, child\_relationship\_code, modified\_by, modified\_date  FROM cp\_signature\_detail cpsd  WHERE cpsd.child\_relationship\_code IS NOT NULL  AND NOT EXISTS (SELECT ref\_data\_code  FROM ref\_data  WHERE domain\_code = ANY ('Relationship', 'Association')  AND ref\_data\_code = cpsd.child\_relationship\_code)  AND EXISTS (SELECT ref\_data\_code  FROM ref\_data  WHERE domain\_code = ANY ('Relationship', 'Association')  AND short\_desc = cpsd.child\_relationship\_code);  /\*  +----------------------------------------------------------------+  | Create a unique audit user for this datafix. |  +----------------------------------------------------------------+  \*/  EXECUTE pkAuditUser.CreateDataFixAudit('13752');  /\*  +----------------------------------------------------------------+  | Correct the case worker signature on the case plan. |  +----------------------------------------------------------------+  \*/  UPDATE cp\_signature\_detail cpsd  SET child\_relationship\_code = (SELECT MIN(ref\_data\_code)  FROM ref\_data  WHERE domain\_code = ANY ('Relationship', 'Association')  AND short\_desc = cpsd.child\_relationship\_code),  modified\_by = pkAuditUser.GetDataFixUser('13752'),  modified\_date = SYSDATE  WHERE cp\_signature\_detail\_id = ANY (SELECT cp\_signature\_detail\_id  FROM datafix\_13752\_cp\_signature)  CREATE TABLE sacwis\_fix.defect\_5928\_fix as (select pr.ROSTER\_ID, pr.ROSTER\_NAMED\_BY\_USER\_ID, pr.MODIFIED\_BY,  pr.MODIFIED\_DATE, pr.CREATED\_BY  from payment\_roster pr where 1 = 2);    DROP TABLE sacwis\_fix.defect\_5928\_disbursement\_fix;  CREATE TABLE sacwis\_fix.defect\_5928\_disbursement\_fix as (select pd.DISBURSEMENT\_ID, pd.CREATED\_BY, pd.MODIFIED\_BY,  pd.MODIFIED\_DATE  from payment\_disbursement pd where 1 = 2);      DECLARE    CURSOR c is select pr.roster\_id, pr.ROSTER\_NAMED\_BY\_USER\_ID, pr.CREATED\_BY, pr.AGENCY\_ID, pr.MODIFIED\_BY, pr.MODIFIED\_DATE,  (select su.SECURITY\_USER\_ID  from security\_user su  where su.PERSON\_ID = pr.ROSTER\_NAMED\_BY\_USER\_ID  and su.security\_user\_id = (select su2.security\_user\_id  from security\_user su2  where su2.PERSON\_ID = pr.ROSTER\_NAMED\_BY\_USER\_ID  and su2.INFO\_SEC\_ID is not null  and su2.start\_date is not null  and rownum = 1  and (to\_char(su2.start\_date, 'yyyymmdd') || su2.security\_user\_id)  = (select max(to\_char(su3.start\_date, 'yyyymmdd') || su3.security\_user\_id)  from security\_user su3  where su3.PERSON\_ID = pr.ROSTER\_NAMED\_BY\_USER\_ID  and su3.INFO\_SEC\_ID is not null  and su3.start\_date is not null  )))  as security\_user\_id\_for\_named ,  (select su.SECURITY\_USER\_ID  from security\_user su  where su.PERSON\_ID = pr.CREATED\_BY  and su.security\_user\_id = (select su2.security\_user\_id  from security\_user su2  where su2.PERSON\_ID = pr.CREATED\_BY  and su2.INFO\_SEC\_ID is not null  and su2.start\_date is not null  and rownum = 1  and (to\_char(su2.start\_date, 'yyyymmdd') || su2.security\_user\_id)  = (select max(to\_char(su3.start\_date, 'yyyymmdd') || su3.security\_user\_id)  from security\_user su3  where su3.PERSON\_ID = pr.CREATED\_BY  and su3.INFO\_SEC\_ID is not null  and su3.start\_date is not null))) as security\_user\_id\_for\_createdby  from payment\_roster pr  left outer join PAYMENT\_DISBURSEMENT pd on pr.DISBURSEMENT\_ID = pd.DISBURSEMENT\_ID  where (pr.created\_by not in (999, 996) or (pr.CREATED\_BY = 999 and nvl(pd.DISBURSED\_STATUS\_FLAG, 0) = 0 ));        CURSOR c1 is select pd.DISBURSEMENT\_ID, pd.CREATED\_BY, pd.AGENCY\_ID, pd.MODIFIED\_BY, pd.MODIFIED\_DATE,  (select su.SECURITY\_USER\_ID  from security\_user su  where su.PERSON\_ID = pd.CREATED\_BY  and su.security\_user\_id = (select su2.security\_user\_id  from security\_user su2  where su2.PERSON\_ID = pd.CREATED\_BY  and su2.INFO\_SEC\_ID is not null  and su2.start\_date is not null  and rownum = 1  and (to\_char(su2.start\_date, 'yyyymmdd') || su2.security\_user\_id)  = (select max(to\_char(su3.start\_date, 'yyyymmdd') || su3.security\_user\_id)  from security\_user su3  where su3.PERSON\_ID = pd.CREATED\_BY  and su3.INFO\_SEC\_ID is not null  and su3.start\_date is not null  )))  as security\_user\_id\_for\_createdby  from payment\_disbursement pd  where pd.created\_by not in (999, 996);    CURSOR employeeWorkHistory (employeeId NUMBER, agencyId NUMBER) is select su.SECURITY\_USER\_ID  from security\_user su,employee\_work\_history ewh  where ewh.employee\_id = employeeId-- su.employee\_id  and ewh.agency\_id = agencyId  and su.EMPLOYEE\_ID = ewh.EMPLOYEE\_ID  and ewh.start\_date = (select max(ewh2.start\_date)  from employee\_work\_history ewh2  where ewh2.employee\_id = employeeId --su.employee\_id  and ewh2.agency\_id = agencyId);        rosterNamedByUserId NUMBER := 0;    rosterCreatedByUserId NUMBER := 0;    disbursementCreatedByUserId NUMBER := 0;    paymentRosterRow employeeWorkHistory%ROWTYPE;    BEGIN    FOR r in c  LOOP    rosterNamedByUserId := r.security\_user\_id\_for\_named;    rosterCreatedByUserId := r.security\_user\_id\_for\_createdby;        open employeeWorkHistory (r.roster\_named\_by\_user\_id, r.agency\_id);  FETCH employeeWorkHistory into paymentRosterRow;  IF(paymentRosterRow.SECURITY\_USER\_ID IS NOT NULL) THEN    rosterNamedByUserId := paymentRosterRow.SECURITY\_USER\_ID;  END IF;    close employeeWorkHistory;    open employeeWorkHistory(r.security\_user\_id\_for\_createdby, r.agency\_id);  FETCH employeeWorkHistory into paymentRosterRow;    IF(paymentRosterRow.SECURITY\_USER\_ID IS NOT NULL) THEN    rosterCreatedByUserId := paymentRosterRow.SECURITY\_USER\_ID;  END IF;      close employeeWorkHistory;    IF((r.security\_user\_id\_for\_named IS NOT NULL )  or (r.security\_user\_id\_for\_createdby IS NOT NULL) ) THEN    insert into sacwis\_fix.defect\_5928\_fix values(r.roster\_id, r.roster\_named\_by\_user\_id, r.modified\_by,  r.modified\_date, r.created\_by);    update payment\_roster pr set pr.roster\_named\_by\_user\_id = rosterNamedByUserId, modified\_by = 999,  modified\_date = SYSDATE, created\_by = rosterCreatedByUserId  where roster\_id = r.roster\_id;    END IF;      END LOOP;  FOR r1 in c1        LOOP  disbursementCreatedByUserId := r1.security\_user\_id\_for\_createdby;      open employeeWorkHistory (r1.security\_user\_id\_for\_createdby, r1.agency\_id);  FETCH employeeWorkHistory into paymentRosterRow;  IF(paymentRosterRow.SECURITY\_USER\_ID IS NOT NULL) THEN    disbursementCreatedByUserId := paymentRosterRow.SECURITY\_USER\_ID;  END IF;    close employeeWorkHistory;      IF(r1.security\_user\_id\_for\_createdby IS NOT NULL) THEN    insert into sacwis\_fix.defect\_5928\_disbursement\_fix values(r1.disbursement\_id, r1.created\_by, r1.modified\_by,  r1.modified\_date);    update payment\_disbursement pd set pd.CREATED\_BY = disbursementCreatedByUserId, pd.MODIFIED\_BY = 999,  pd.MODIFIED\_DATE = SYSDATE  where pd.DISBURSEMENT\_ID = r1.disbursement\_id;  END IF;  END LOOP |
| Audit Tables Updated | New Tables and Modified Tables  Reference Above for Table and Column Adds and Modifications |
| List of reports deployed : | Reference Bill Ennis |
| List of reports deactivated : | Reference Bill Ennis |

**Section 3 – Testing Plan/Results**

3.1 Test Plan

|  |  |
| --- | --- |
| Identify the database used to test the software/SQL: | Flair , system test |
| How many days removed from production is the data in this database? | n/a |
| Name of person(s) completing the test in this database: | Stephen Fibelkorn |
| Identify the database used for user acceptance test: | UAT |
| How many days removed from production is the data in this database? | Start of build Cycle (32 Days) |
| Name of person(s) completing the user acceptance test in this database: | Iroabuchi Arum |

**Section 4 – Production Execution**

* 1. Execution Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identify whether a full or partial backup of the production database is needed, or indicate that no backup is needed. | Full | **Successful RMAN backup Needed** | Partial |  |
| Explanation: | | | |
| Identify when the backup will be taken: | Date: 09/15/2011  Time: 9.00 pm | | | |
| Identify who will apply the software/SQL: | ODJFS DBA | | | |

* 1. Execution Results

|  |  |
| --- | --- |
| Location of results documentation: | Results to be e-mailed to the SACWIS team. |