Mapping the OC Validation and Derivation Procedure Interface to their RDBMS Components: Understanding the Internals of Procedures

Sunil G. Singh

Introduction

- Sunil G. Singh of DBMS Consulting, Inc.
- Specialize in Oracle Pharmaceutical and E-Business implementations and long-term support.
- Additional copies of presentation available at Booths 19-20, as well as time for follow-up questions.

Acknowledgements

- Thanks to the OCUG for this opportunity to speak.
- Special thanks to Kay Wilson and Jane Kuczma of the OCUG-Validation and Derivation Procedures Focus Group for their nearly infinite patience and gracious acceptance of this paper.

Goals

- Explanation of how validation and derivation procedures are mapped into OC using a simple validation procedure from the OC QA Smoke Test
- Describe a method for mapping all parts of functionality to their database equivalents.
- Technical discussion, not a functional approach

What are Validation and Derivation Procedures?

- Rules or “edit checks” for a study, usually for multivariate checks.
- In reality, each validation and derivation procedures is actually a PL/SQL package in the RXC_PD schema.
- These packages are executed individually or at Batch Validation time.

Deriving the Existing Code for Validation Procedures

- Since all procedures are actually PL/SQL packages in the RDBMS, all of the code for **complied** procedures is in sys.source$, which is visible from DBA_SOURCE, ALL_SOURCE (if the account has all privileges on the PL/SQL package) and USER_SOURCE (as the user RXC_PD).
- The package name comes from the RXC.PROCEDURES table, from the PROCEDURE_ID and the PROCEDURE_VER_SN columns
- The conditions of the procedure are stored in the RXC.PROCEDURE_DETAILS
- The name of the package is RXC_PD.RXCPD_<PROCEDURE_ID>_<PROCEDURE_VER_SN>

Deriving the Existing Code for Validation Procedures (2)

- Therefore, once the PROCEDURE_ID and the PROCEDURE_VER_SN columns are known, the package header can be drawn out of the RDBMS using SQL*Plus:
  - select clinical_study_id, study from clinical_studies where study = ‘<study_name>';
  - select name, procedure_id, procedure_ver_sn from procedures where clinical_study_id = 111 and name = ‘<procedure’s_name>‘;
  - set echo off feedback off termout on pagesize 0 linesize 255 trimspool on
  - spool /tmp/rxcpd_<procedure_id>_<procedure_ver_sn>_head.sql
  - select substr(text,1,255) from dba_source where name = 'RXCPD_<procedure_id>_<procedure_ver_sn>' and owner = 'RXC_PD' and type = 'PACKAGE' order by line;

rxcpd_11_0_head - Notepad

File Edit Search Help

```
package RXCPD_11_0 as

/* cursor for getting DEMOG2 Production */
cursor D_CUR(
I_PATIENT_POSITION_ID in RECEIVED_DCMS.PATIENT_POSITION_ID%TYPE,
I_BEGIN_VISIT_NUMBER  in RECEIVED_DCMS.VISIT_NUMBER%TYPE := RXCPDSTD.C_BEGIN_VISIT_NUMBER,
I_END_VISIT_NUMBER    in RECEIVED_DCMS.VISIT_NUMBER%TYPE := RXCPDSTD.C_END_VISIT_NUMBER) is
    select /*+ ordered use_nl(RDCM RES)
           index(RDCM RECEIVED_DCM_UK2_IDX) */
           RDCM.RECEIVED_DCM_ID,
           RDCM.RECEIVED_DCM_ENTRY_TS,
           RDCM.INVESTIGATOR_ID,
           RDCM.SITE_ID,
           RDCM.DCM_ID,
           RDCM.DCM_SUBSET_SN,
           RDCM.DCM_DATE,
           RDCM.DCM_TIME,
           RDCM.ACTUAL_EVENT_ID,
           RDCM.LAB_ID,
           RDCM.LAB LAB,
           RDCM.LAB_RANGE_SUBSET_NUM,
           RDCM.QUALIFYING_VALUE,
           RDCM.SUBEVENT_NUMBER,
           RDCM.CLIN_PLAN_EVE_ID,
           RDCM.CLIN_PLAN_EVE_NAME,
           RDCM.VISIT_NUMBER,
           RES.REPEAT_SN,
max(decode(RES.DCM_QUESTION_ID,11,substrb(RES.VALUE_TEXT,1,1),null)) SEX2,
max(decode(RES.DCM_QUESTION_ID,11,substrb(RES.EXCEPTION_VALUE_TEXT,1,20),NULL)) SEX2$EXC_VAL,
max(decode(RES.DCM_QUESTION_ID,11,RES.RESPONSE_ID,NULL)) SEX2$RESP_ID,
max(decode(RES.DCM_QUESTION_ID,11,RES.RESPONSE_ENTRY_TS,NULL)) SEX2$ENT_TS,
max(decode(RES.DCM_QUESTION_ID,111,to_number(RES.VALUE_TEXT),null)) WEIGHT2,
max(decode(RES.DCM_QUESTION_ID,111,substrb(RES.EXCEPTION_VALUE_TEXT,1,20),NULL)) WEIGHT2$EXC_VAL,
max(decode(RES.DCM_QUESTION_ID,111,RES.RESPONSE_ID,NULL)) WEIGHT2$RESP_ID,
max(decode(RES.DCM_QUESTION_ID,111,RES.RESPONSE_ENTRY_TS,NULL)) WEIGHT2$ENT_TS
      from RECEIVED_DCMS RDCM,
           RESPONSES RES
     where RDCM.PATIENT_POSITION_ID = I_PATIENT_POSITION_ID
       and RDCM.DCM_ID = 11
       and RDCM.END_TS = TO_DATE(3000000,'J')
       and RDCM.RECEIVED_DCM_ID+0 = RES.RECEIVED_DCM_ID
       and RES.END_TS = TO_DATE(3000000,'J')
       and RES.DCM_QUESTION_GROUP_ID = 11 and RES.CLINICAL_STUDY_ID = 111
       and RDCM.ACCESSIBLE_TS <= SYSDATE
       and RDCM.VISIT_NUMBER between I_BEGIN_VISIT_NUMBER and I_END_VISIT_NUMBER
       and res.dcm_question_id in (
       11, 111)
  group by RDCM.RECEIVED_DCM_ID,
```

rxcpd_11_0_head - Notepad

File Edit Search Help

```
  group by RDCM.RECEIVED_DCM_ID,
           RDCM.RECEIVED_DCM_ENTRY_TS,
           RDCM.INVESTIGATOR_ID,
           RDCM.SITE_ID,
           RDCM.DCM_ID,
           RDCM.DCM_SUBSET_SN,
           RDCM.DCM_DATE,
           RDCM.DCM_TIME,
           RDCM.ACTUAL_EVENT_ID,
           RDCM.LAB_ID,
           RDCM.LAB,
           RDCM.LAB_RANGE_SUBSET_NUM,
           RDCM.QUALIFYING_VALUE,
           RDCM.SUBEVENT_NUMBER,
           RDCM.CLIN_PLAN_EVE_ID,
           RDCM.CLIN_PLAN_EVE_NAME,
           RDCM.VISIT_NUMBER,
           RES.REPEAT_SN
ORDER BY RDCM.VISIT_NUMBER ASC,RDCM.SUBEVENT_NUMBER ASC, RES.REPEAT_SN ASC;
D D_CUR%rowtype;
D$BEGIN_SEQNUM  number := -99999;
D$END_SEQNUM  number := 99999;

/* cursor for getting DEMOG2 Test */
cursor D_CURT(
I_PATIENT_POSITION_ID in RECEIVED_DCMS.PATIENT_POSITION_ID%TYPE,
I_BEGIN_VISIT_NUMBER  in RECEIVED_DCMS.VISIT_NUMBER%TYPE := RXCPDSTD.C_BEGIN_VISIT_NUMBER,
I_END_VISIT_NUMBER    in RECEIVED_DCMS.VISIT_NUMBER%TYPE := RXCPDSTD.C_END_VISIT_NUMBER) is
    select /*+ ordered use_nl(RDCM RES)
           index(RDCM RECEIVED_DCM_UK2_IDX) */
           RDCM.RECEIVED_DCM_ID,
           RDCM.RECEIVED_DCM_ENTRY_TS,
           RDCM.INVESTIGATOR_ID,
           RDCM.SITE_ID,
           RDCM.DCM_ID,
           RDCM.DCM_SUBSET_SN,
           RDCM.DCM_DATE,
           RDCM.DCM_TIME,
           RDCM.ACTUAL_EVENT_ID,
           RDCM.LAB_ID,
           RDCM.LAB LAB,
           RDCM.LAB_RANGE_SUBSET_NUM,
           RDCM.QUALIFYING_VALUE,
           RDCM.SUBEVENT_NUMBER,
           RDCM.CLIN_PLAN_EVE_ID,
           RDCM.CLIN_PLAN_EVE_NAME,
           RDCM.VISIT_NUMBER,
           RES.REPEAT_SN,
```

rxcpd_11_0_head - Notepad

File Edit Search Help

```
           RES.REPEAT_SN,
max(decode(RES.DCM_QUESTION_ID,11,substrb(RES.VALUE_TEXT,1,1),null)) SEX2,
max(decode(RES.DCM_QUESTION_ID,11,substrb(RES.EXCEPTION_VALUE_TEXT,1,20),NULL)) SEX2$EXC_VAL,
max(decode(RES.DCM_QUESTION_ID,11,RES.RESPONSE_ID,NULL)) SEX2$RESP_ID,
max(decode(RES.DCM_QUESTION_ID,11,RES.RESPONSE_ENTRY_TS,NULL)) SEX2$ENT_TS,
max(decode(RES.DCM_QUESTION_ID,111,to_number(RES.VALUE_TEXT),null)) WEIGHT2,
max(decode(RES.DCM_QUESTION_ID,111,substrb(RES.EXCEPTION_VALUE_TEXT,1,20),NULL)) WEIGHT2$EXC_VAL,
max(decode(RES.DCM_QUESTION_ID,111,RES.RESPONSE_ID,NULL)) WEIGHT2$RESP_ID,
max(decode(RES.DCM_QUESTION_ID,111,RES.RESPONSE_ENTRY_TS,NULL)) WEIGHT2$ENT_TS
      from RECEIVED_DCMST RDCM,
           RESPONSEST     RES
     where RDCM.PATIENT_POSITION_ID = I_PATIENT_POSITION_ID
       and RDCM.DCM_ID = 11
       and RDCM.END_TS = TO_DATE(3000000,'J')
       and RDCM.RECEIVED_DCM_ID+0 = RES.RECEIVED_DCM_ID
       and RES.END_TS = TO_DATE(3000000,'J')
       and RES.DCM_QUESTION_GROUP_ID = 11 and RES.CLINICAL_STUDY_ID = 111
       and RDCM.ACCESSIBLE_TS <= SYSDATE
       and RDCM.VISIT_NUMBER between I_BEGIN_VISIT_NUMBER AND I_END_VISIT_NUMBER
       and res.dcm_question_id in (
       11, 111)
  group by RDCM.RECEIVED_DCM_ID,
           RDCM.RECEIVED_DCM_ENTRY_TS,
           RDCM.INVESTIGATOR_ID,
           RDCM.SITE_ID,
           RDCM.DCM_ID,
           RDCM.DCM_SUBSET_SN,
           RDCM.DCM_DATE,
           RDCM.DCM_TIME,
           RDCM.ACTUAL_EVENT_ID,
           RDCM.LAB_ID,
           RDCM.LAB,
           RDCM.LAB_RANGE_SUBSET_NUM,
           RDCM.QUALIFYING_VALUE,
           RDCM.SUBEVENT_NUMBER,
           RDCM.CLIN_PLAN_EVE_ID,
           RDCM.CLIN_PLAN_EVE_NAME,
           RDCM.VISIT_NUMBER,
           RES.REPEAT_SN
ORDER BY RDCM.VISIT_NUMBER ASC,RDCM.SUBEVENT_NUMBER ASC, RES.REPEAT_SN ASC;
/*--------------| Table Declarations |---------------------------------*/
DET1_DISC_TAB    RXCPDSTD.CHECK_DISC_TAB_TYPE;
/*--------------| Procedure and Function Declarations |----------------*/

procedure MAIN (
I_CLINICAL_STUDY_ID         in PATIENT_POSITIONS.CLINICAL_STUDY_ID%TYPE,
I_CLINICAL_STUDY_VERSION_ID in PATIENT_POSITIONS.CLINICAL_STUDY_VERSION_ID%TYPE,
I_DATA_MODIFIED_FLAG        in PATIENT_POSITIONS.DATA_MODIFIED_FLAG%TYPE,
```

rxcpd_11_0_head - Notepad

File Edit Search Help

```
  group by RDCM.RECEIVED_DCM_ID,
           RDCM.RECEIVED_DCM_ENTRY_TS,
           RDCM.INVESTIGATOR_ID,
           RDCM.SITE_ID,
           RDCM.DCM_ID,
           RDCM.DCM_SUBSET_SN,
           RDCM.DCM_DATE,
           RDCM.DCM_TIME,
           RDCM.ACTUAL_EVENT_ID,
           RDCM.LAB_ID,
           RDCM.LAB,
           RDCM.LAB_RANGE_SUBSET_NUM,
           RDCM.QUALIFYING_VALUE,
           RDCM.SUBEVENT_NUMBER,
           RDCM.CLIN_PLAN_EVE_ID,
           RDCM.CLIN_PLAN_EVE_NAME,
           RDCM.VISIT_NUMBER,
           RES.REPEAT_SN
ORDER BY RDCM.VISIT_NUMBER ASC,RDCM.SUBEVENT_NUMBER ASC, RES.REPEAT_SN ASC;
/*--------------| Table Declarations |---------------------------------*/
DET1_DISC_TAB    RXCPDSTD.CHECK_DISC_TAB_TYPE;
/*--------------| Procedure and Function Declarations |----------------*/

procedure MAIN (
I_CLINICAL_STUDY_ID         in PATIENT_POSITIONS.CLINICAL_STUDY_ID%TYPE,
I_CLINICAL_STUDY_VERSION_ID in PATIENT_POSITIONS.CLINICAL_STUDY_VERSION_ID%TYPE,
I_DATA_MODIFIED_FLAG        in PATIENT_POSITIONS.DATA_MODIFIED_FLAG%TYPE,
I_PROCEDURE_ID              in PROCEDURES.PROCEDURE_ID%TYPE,
I_PROCEDURE_VERSION_SN      in PROCEDURES.PROCEDURE_VER_SN%TYPE,
I_PROCEDURE_TYPE_CODE       in PROCEDURES.PROCEDURE_TYPE_CODE%TYPE,
I_USERNAME                  in varchar2,
I_DEBUG                     in varchar2,
I_CURRENT_BATCH_TS          in varchar2,
I_LAST_BATCH_TS             in varchar2,
I_CURRENT_LOCATION          in PATIENT_POSITIONS.OWNING_LOCATION%TYPE,
I_MODE                      in varchar2,
I_TIMER                     in varchar2 := 'N',
I_LAB_DEPENDENT_FLAG        in PROCEDURES.LAB_DEPENDENT_FLAG%TYPE);


procedure INSERT_DISCREPANCY (
I_REVIEW_STATUS             in PROCEDURE_DETAILS.INIT_DISCR_REVIEW_STATUS_CODE%TYPE,
I_PROCEDURE_DETAIL_ID       in PROCEDURE_DETAILS.PROCEDURE_DETAIL_ID%TYPE);

procedure EXCEPTION_HANDLING (
I_ERR_MSG                   in varchar2);

END RXCPD_11_0;
```

Deriving the Existing Code for Validation Procedures (3)

- Similarly, the the package body can be drawn out of the RDBMS using SQL*Plus:
  - select clinical_study_id, study from clinical_studies where study = ‘<study_name>';
  - select name, procedure_id, procedure_ver_sn from procedures where clinical_study_id = 111 and name = ‘<procedure’s_name>‘;
  - set echo off feedback off termout on pagesize 0 linesize 255 trimspool on
  - spool /tmp/rxcpd_<procedure_id>_<procedure_ver_sn>_body.sql
  - select substr(text,1,255) from dba_source where name = 'RXCPD_<procedure_id>_<procedure_ver_sn>' and owner = 'RXC_PD' and type = 'PACKAGE BODY' order by line;

rxcpd_11_0_body - Notepad

File Edit Search Help

```
package body RXCPD_11_0 as

procedure MAIN /*---------------------------------------*/(
   I_CLINICAL_STUDY_ID         in PATIENT_POSITIONS.CLINICAL_STUDY_ID%TYPE,
   I_CLINICAL_STUDY_VERSION_ID in PATIENT_POSITIONS.CLINICAL_STUDY_VERSION_ID%TYPE,
   I_DATA_MODIFIED_FLAG        in PATIENT_POSITIONS.DATA_MODIFIED_FLAG%TYPE,
   I_PROCEDURE_ID              in PROCEDURES.PROCEDURE_ID%TYPE,
   I_PROCEDURE_VERSION_SN      in PROCEDURES.PROCEDURE_VER_SN%TYPE,
   I_PROCEDURE_TYPE_CODE       in PROCEDURES.PROCEDURE_TYPE_CODE%TYPE,
   I_USERNAME                  in varchar2,
   I_DEBUG                     in varchar2,
   I_CURRENT_BATCH_TS          in varchar2,
   I_LAST_BATCH_TS             in varchar2,
   I_CURRENT_LOCATION          in PATIENT_POSITIONS.OWNING_LOCATION%TYPE,
   I_MODE                      in varchar2,
   I_TIMER                     in varchar2 := 'N',
   I_LAB_DEPENDENT_FLAG        in PROCEDURES.LAB_DEPENDENT_FLAG%TYPE ) is


   V_DUPLICATE_DISCREPANCY     boolean;
   CORREL_EVENT_ID             number(10);
/****** Declaration ******/

/****** Declaration ******/

begin

   if I_TIMER = 'Y' then RXCPDSTD.CAPTURE_TIME; end if;


   RXCPDSTD.V_CLINICAL_STUDY_ID         := I_CLINICAL_STUDY_ID;
   RXCPDSTD.V_CLINICAL_STUDY_VERSION_ID := I_CLINICAL_STUDY_VERSION_ID;
   RXCPDSTD.V_DATA_MODIFIED_FLAG        := I_DATA_MODIFIED_FLAG;
   RXCPDSTD.V_PROCEDURE_ID              := I_PROCEDURE_ID;
   RXCPDSTD.V_PROCEDURE_VERSION_SN      := I_PROCEDURE_VERSION_SN;
   RXCPDSTD.V_PROCEDURE_TYPE_CODE       := I_PROCEDURE_TYPE_CODE;
   RXCPDSTD.V_USERNAME                  := I_USERNAME;
   RXCPDSTD.V_DEBUG                     := I_DEBUG;
   RXCPDSTD.V_CURRENT_BATCH_TS          := to_date(I_CURRENT_BATCH_TS,'DD-MON-YYYY HH24:MI:SS');
   RXCPDSTD.V_LAST_BATCH_TS             := to_date(I_LAST_BATCH_TS,'DD-MON-YYYY HH24:MI:SS');
   RXCPDSTD.V_CURRENT_LOCATION          := I_CURRENT_LOCATION;
   RXCPDSTD.V_MODE                      := I_MODE;
   RXCPDSTD.V_LAB_DEPENDENT_FLAG        := I_LAB_DEPENDENT_FLAG;


   RXCPDSTD.V_CODE_LOCATION := 'Main Begin';
   /****** Main Begin ******/
```

rxcpd_11_0_body - Notepad

File Edit Search Help

```
/****** Main Begin ******/
if I_MODE = 'P'
  then open RXCPDSTD.PATIENTS_CUR;  /* Production */
  else open RXCPDSTD.PATIENTS_CURT; /* Test */
end if;
loop <<fetch_next_patient>>
   if I_MODE = 'P'
     then  /* Production */
       fetch RXCPDSTD.PATIENTS_CUR into RXCPDSTD.PATIENTS_REC;
       exit when RXCPDSTD.PATIENTS_CUR%notfound;
     else  /* Test */
       fetch RXCPDSTD.PATIENTS_CURT into RXCPDSTD.PATIENTS_REC;
       exit when RXCPDSTD.PATIENTS_CURT%notfound;
   end if;
   if RXCPDSTD.V_DEBUG = 'Y' then
     dbms_output.put_line('------------------------------');
     dbms_output.put_line(
       'Processing Patient: '||RXCPDSTD.PATIENTS_REC.PATIENT);
   end if;
   begin
      RXCPDSTD.V_CODE_LOCATION := 'Post Patient';
      /****** Post Patient ******/

      /****** Post Patient ******/
      if I_MODE = 'P'
        then open D_CUR(RXCPDSTD.PATIENTS_REC.PATIENT_POSITION_ID, D$BEGIN_SEQNUM, D$END_SEQNUM);
        else open D_CURT(RXCPDSTD.PATIENTS_REC.PATIENT_POSITION_ID, D$BEGIN_SEQNUM, D$END_SEQNUM);
      end if;
      loop << fetch_D_cur>>
         RXCPDSTD.V_CODE_LOCATION := 'Fetching D data';
         if I_MODE = 'P'
           then fetch D_CUR into D;
           else fetch D_CURT into D;
         end if;
         if I_MODE = 'P'
           then exit when D_CUR%notfound;
           else exit when D_CURT%notfound;
         end if;

         RXCPDSTD.V_CODE_LOCATION := 'Post QG 11';
         /****** Post QG D ******/

         /****** Post QG D ******/
         if RXCPDSTD.V_DEBUG = 'Y' then
            RXCPDSTD.PRINT_DCM_HEADER('D',D.VISIT_NUMBER,D.DCM_DATE,D.REPEAT_SN);
            DBMS_OUTPUT.PUT_LINE('D.SEX2 = '||D.SEX2);
            DBMS_OUTPUT.PUT_LINE('D.WEIGHT2 = '||D.WEIGHT2);
         end if;
```

rxcpd_11_0_body - Notepad

File Edit Search Help

```
            RXCPDSTD.V_CODE_LOCATION := 'Pre Details';
            /****** Pre Details ******/

            /****** Pre Details ******/
            <<detail_1>>
               RXCPDSTD.V_CODE_LOCATION := 'Detail 1 - Sex Dependant Weight Check for Oracle QA Study 2';
               if ((D.SEX2='M' and D.WEIGHT2 NOT BETWEEN 90 and 350) OR (D.SEX2='F' and D.WEIGHT2 NOT BETWEEN 70 and 330))
then
                  if RXCPDSTD.V_DEBUG = 'Y' then DBMS_OUTPUT.PUT_LINE('Discrepancy found for '||RXCPDSTD.V_CODE_LOCATION);
end if;
                  DET1_DISC_TAB(1).RESPONSE_ID := D.SEX2$RESP_ID;
                  DET1_DISC_TAB(1).RESPONSE_ENTRY_TS := D.SEX2$ENT_TS;
                  DET1_DISC_TAB(2).RESPONSE_ID := D.WEIGHT2$RESP_ID;
                  DET1_DISC_TAB(2).RESPONSE_ENTRY_TS := D.WEIGHT2$ENT_TS;
                  RXCPDSTD.CHECK_DISCREPANCY (11,D.received_dcm_id,DET1_DISC_TAB,V_DUPLICATE_DISCREPANCY);
                  if not V_DUPLICATE_DISCREPANCY then
                     INSERT_DISCREPANCY('UNREVIEWED',11);
                     RXCPDSTD.INSERT_VRV('D.SEX2',D.SEX2$resp_id,D.SEX2$ent_ts,D.SEX2);
                     RXCPDSTD.INSERT_VRV('D.WEIGHT2',D.WEIGHT2$resp_id,D.WEIGHT2$ent_ts,D.WEIGHT2);
                     RXCPDSTD.STORE_DISCREPANCY_MESSAGE;
                  end if;
                  GOTO no_more_details;
               end if;
            <<no_more_details>>
            RXCPDSTD.V_CODE_LOCATION := 'Post Details';
            /****** Post Details ******/

            /****** Post Details ******/
         end loop; /* D cursor loop */
         if I_MODE = 'P' then close D_CUR; else close D_CURT; end if;
      end;
   end loop; /* patients cursor loop */
   if I_MODE = 'P'
     then close rxcpdstd.patients_cur;
     else close rxcpdstd.patients_curt;
   end if;
   RXCPDSTD.V_CODE_LOCATION := 'Main End';
   /****** Main End ******/

   /****** Main End ******/
     if I_TIMER = 'Y' then RXCPDSTD.SHOW_ELAPSED_TIME; end if;
     exception when others then EXCEPTION_HANDLING(sqlerrm);
end MAIN; /* end of procedure */

procedure INSERT_DISCREPANCY /* ------------------------------------ */
(I_REVIEW_STATUS       in PROCEDURE_DETAILS.INIT_DISCR_REVIEW_STATUS_CODE%type,
 I_PROCEDURE_DETAIL_ID in PROCEDURE_DETAILS.PROCEDURE_DETAIL_ID%type) is
begin
```

rxcpd_11_0_body - Notepad

File Edit Search Help

```
         if I_MODE = 'P' then close D_CUR; else close D_CURT; end if;
      end;
   end loop; /* patients cursor loop */
   if I_MODE = 'P'
     then close rxcpdstd.patients_cur;
     else close rxcpdstd.patients_curt;
   end if;
   RXCPDSTD.V_CODE_LOCATION := 'Main End';
   /****** Main End ******/

   /****** Main End ******/
     if I_TIMER = 'Y' then RXCPDSTD.SHOW_ELAPSED_TIME; end if;
     exception when others then EXCEPTION_HANDLING(sqlerrm);
end MAIN; /* end of procedure */

procedure INSERT_DISCREPANCY /* ------------------------------------ */
(I_REVIEW_STATUS       in PROCEDURE_DETAILS.INIT_DISCR_REVIEW_STATUS_CODE%type,
 I_PROCEDURE_DETAIL_ID in PROCEDURE_DETAILS.PROCEDURE_DETAIL_ID%type) is
begin
  RXCPDSTD.V_CODE_LOCATION := 'Insert Discrepancy Wrapper';
  RXCPDSTD.INSERT_DISCREPANCY (D.INVESTIGATOR_ID,D.SITE_ID,I_REVIEW_STATUS,
     D.RECEIVED_DCM_ID,D.RECEIVED_DCM_ENTRY_TS,I_PROCEDURE_DETAIL_ID,
     D.CLIN_PLAN_EVE_ID,D.ACTUAL_EVENT_ID,
     D.DCM_ID,D.SUBEVENT_NUMBER);
  exception when others then EXCEPTION_HANDLING(sqlerrm);
end INSERT_DISCREPANCY;

procedure EXCEPTION_HANDLING /* ------------------------------------ */
(I_ERR_MSG             in varchar2) is
begin
  if RXCPDSTD.V_MODE = 'P'
    then close D_CUR;
    else close D_CURT;
  end if;
  RXCPDSTD.EXCEPTION_HANDLING(I_ERR_MSG);
  exception when others then RXCPDSTD.EXCEPTION_HANDLING(sqlerrm);
end EXCEPTION_HANDLING;


/* --------------| Package initialization Section |-------------------*/
begin
  RXCPDSTD.V_CODE_LOCATION := 'Package Initialization';

  /* Populate discrepancy check table */
  select 'Q','D.SEX2',null,null,null into DET1_DISC_TAB(1) from dual;
  select 'Q','D.WEIGHT2',null,null,null into DET1_DISC_TAB(2) from dual;
  exception when others then RXCPDSTD.EXCEPTION_HANDLING(sqlerrm);
end RXCPD_11_0;
```

Analysis of the Package (1)

- Understanding the package header and package body of the Validation and Derivation procedure is the key to
  - Writing custom code for the package
  - Debugging procedures
  - Performance tuning and tracing
- Examining package from main
- Open the rxcpdstd.patients_cur
  - Rxcpdstd is the core package on which all procedures are based.

rxcpdstd_patients_cur - Notepad

File Edit Search Help

```
cursor PATIENTS_CUR is /* production */
/* Add branch for RDC 3.2. Get all patients with modified data for a particular site. */
select /*+ ORDERED */
       PAPO.PATIENT_POSITION_ID, PAPO.CLINICAL_STUDY_ID, PAPO.REPORTED_SEX,
       PAPO.REPORTED_BIRTH_DATE, PAPO.PATIENT, PAPO.EARLY_TERMINATION_FLAG,
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
       PAPO.INCLUSION_EXCLUSION_DATE, PAPO.REPORTED_PATIENT_REFERENCE,
       PAPO.REPORTED_INITIALS, PAPO.REPORTED_DATE_LAST_PREGNANCY,
       PAPO.REPORTED_DEATH_DATE, PAPO.FIRST_SCREENING_DATE,
       PAPO.TERMINATION_DATE, -999999999, -999999999,
       ocl.patient_pad(PAPO.PATIENT) PATIENT_ORDER
  from RXA_DES.STUDY_SITE_PATIENT_POSITIONS SSPP,
       RXA_DES.PATIENT_POSITIONS PAPO,
       RXA_DES.PATIENT_DM_TRACKING PTDT
 where SSPP.CLINICAL_STUDY_ID = V_CLINICAL_STUDY_ID
   and SSPP.SITE_ID = V_SITE_ID
   and SSPP.PATIENT_POSITION_ID = PAPO.PATIENT_POSITION_ID
   and PAPO.HAS_DATA_FLAG = 'Y'
   and PAPO.OWNING_LOCATION = V_CURRENT_LOCATION
   and PAPO.FREEZE_FLAG = 'N'
   and PAPO.PATIENT_POSITION_ID = PTDT.PATIENT_POSITION_ID
   /* patients with modified data only */
   and (PAPO.DATA_MODIFIED_FLAG = 'Y' or PTDT.LOCAL_DATA_MODIFIED_FLAG = 'Y')
   and V_TARGET_PATIENT = -1
UNION ALL
/* Change over for RDC 3.2 branch */
select PAPO.PATIENT_POSITION_ID, PAPO.CLINICAL_STUDY_ID, PAPO.REPORTED_SEX,
       PAPO.REPORTED_BIRTH_DATE, PAPO.PATIENT, PAPO.EARLY_TERMINATION_FLAG,
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
       PAPO.INCLUSION_EXCLUSION_DATE, PAPO.REPORTED_PATIENT_REFERENCE,
       PAPO.REPORTED_INITIALS, PAPO.REPORTED_DATE_LAST_PREGNANCY,
       PAPO.REPORTED_DEATH_DATE, PAPO.FIRST_SCREENING_DATE,
       PAPO.TERMINATION_DATE, -999999999, -999999999,
       ocl.patient_pad(PAPO.PATIENT) PATIENT_ORDER
  from RXA_DES.PATIENT_POSITIONS PAPO
 where PAPO.CLINICAL_STUDY_ID = V_CLINICAL_STUDY_ID
   and PAPO.CLINICAL_STUDY_VERSION_ID = V_CLINICAL_STUDY_VERSION_ID
   and PAPO.HAS_DATA_FLAG = 'Y'
   /* patients with modified data only (batch validation) */
   and (V_DATA_MODIFIED_FLAG = 'Y' and PAPO.DATA_MODIFIED_FLAG = 'Y')
   and PAPO.OWNING_LOCATION = V_CURRENT_LOCATION
   and PAPO.FREEZE_FLAG = 'N'
   and V_LAB_DEPENDENT_FLAG = 'N'
   and V_TARGET_PATIENT = 0
UNION ALL
select PAPO.PATIENT_POSITION_ID, PAPO.CLINICAL_STUDY_ID, PAPO.REPORTED_SEX,
       PAPO.REPORTED_BIRTH_DATE, PAPO.PATIENT, PAPO.EARLY_TERMINATION_FLAG,
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
```

rxcpdstd_patients_cur - Notepad

File Edit Search Help

```
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
       PAPO.INCLUSION_EXCLUSION_DATE, PAPO.REPORTED_PATIENT_REFERENCE,
       PAPO.REPORTED_INITIALS, PAPO.REPORTED_DATE_LAST_PREGNANCY,
       PAPO.REPORTED_DEATH_DATE, PAPO.FIRST_SCREENING_DATE,
       PAPO.TERMINATION_DATE, -999999999, -999999999,
       ocl.patient_pad(PAPO.PATIENT) PATIENT_ORDER
  from RXA_DES.PATIENT_POSITIONS PAPO
 where PAPO.CLINICAL_STUDY_ID = V_CLINICAL_STUDY_ID
   and PAPO.CLINICAL_STUDY_VERSION_ID = V_CLINICAL_STUDY_VERSION_ID
   and PAPO.HAS_DATA_FLAG = 'Y'
   /* all patients */
   and V_DATA_MODIFIED_FLAG = 'N'
   and PAPO.OWNING_LOCATION = V_CURRENT_LOCATION
   and PAPO.FREEZE_FLAG = decode(PAPO.FREEZE_FLAG,'N','N',V_IS_DERIVATION_FLAG) /* Bug 1177342 */
   and V_TARGET_PATIENT = 0
UNION ALL
select PAPO.PATIENT_POSITION_ID, PAPO.CLINICAL_STUDY_ID, PAPO.REPORTED_SEX,
       PAPO.REPORTED_BIRTH_DATE, PAPO.PATIENT, PAPO.EARLY_TERMINATION_FLAG,
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
       PAPO.INCLUSION_EXCLUSION_DATE, PAPO.REPORTED_PATIENT_REFERENCE,
       PAPO.REPORTED_INITIALS, PAPO.REPORTED_DATE_LAST_PREGNANCY,
       PAPO.REPORTED_DEATH_DATE, PAPO.FIRST_SCREENING_DATE,
       PAPO.TERMINATION_DATE, -999999999, -999999999,
       ocl.patient_pad(PAPO.PATIENT) PATIENT_ORDER
  from RXA_DES.PATIENT_POSITIONS PAPO
 where PAPO.CLINICAL_STUDY_ID = V_CLINICAL_STUDY_ID
   and PAPO.CLINICAL_STUDY_VERSION_ID = V_CLINICAL_STUDY_VERSION_ID
   and PAPO.HAS_DATA_FLAG = 'Y'
   and V_DATA_MODIFIED_FLAG = 'Y'
   /* patients with modified data and/or affected by lab changes */
   and (V_LAB_DEPENDENT_FLAG = 'Y'
       and ((PAPO.LAST_LAB_MODIFICATION_TS > PAPO.LAST_LAB_BATCH_TS) or
            (PAPO.DATA_MODIFIED_FLAG = 'Y')))
   and PAPO.OWNING_LOCATION = V_CURRENT_LOCATION
   and PAPO.FREEZE_FLAG = 'N'
   and V_TARGET_PATIENT = 0
UNION ALL
select PAPO.PATIENT_POSITION_ID, PAPO.CLINICAL_STUDY_ID, PAPO.REPORTED_SEX,
       PAPO.REPORTED_BIRTH_DATE, PAPO.PATIENT, PAPO.EARLY_TERMINATION_FLAG,
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
       PAPO.INCLUSION_EXCLUSION_DATE, PAPO.REPORTED_PATIENT_REFERENCE,
       PAPO.REPORTED_INITIALS, PAPO.REPORTED_DATE_LAST_PREGNANCY,
       PAPO.REPORTED_DEATH_DATE, PAPO.FIRST_SCREENING_DATE,
       PAPO.TERMINATION_DATE, -999999999, -999999999,
       ocl.patient_pad(PAPO.PATIENT) PATIENT_ORDER
  from RXA_DES.PATIENT_POSITIONS PAPO
 where PAPO.CLINICAL_STUDY_ID = V_CLINICAL_STUDY_ID
   and PAPO.CLINICAL_STUDY_VERSION_ID = V_CLINICAL_STUDY_VERSION_ID
```

rxcpdstd_patients_cur - Notepad

File Edit Search Help

```
   and PAPO.CLINICAL_STUDY_VERSION_ID = V_CLINICAL_STUDY_VERSION_ID
   and PAPO.HAS_DATA_FLAG = 'Y'
   /* all patients */
   and V_DATA_MODIFIED_FLAG = 'N'
   and PAPO.OWNING_LOCATION = V_CURRENT_LOCATION
   and PAPO.FREEZE_FLAG = decode(PAPO.FREEZE_FLAG,'N','N',V_IS_DERIVATION_FLAG) /* Bug 1177342 */
   and V_TARGET_PATIENT = 0
UNION ALL
select PAPO.PATIENT_POSITION_ID, PAPO.CLINICAL_STUDY_ID, PAPO.REPORTED_SEX,
       PAPO.REPORTED_BIRTH_DATE, PAPO.PATIENT, PAPO.EARLY_TERMINATION_FLAG,
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
       PAPO.INCLUSION_EXCLUSION_DATE, PAPO.REPORTED_PATIENT_REFERENCE,
       PAPO.REPORTED_INITIALS, PAPO.REPORTED_DATE_LAST_PREGNANCY,
       PAPO.REPORTED_DEATH_DATE, PAPO.FIRST_SCREENING_DATE,
       PAPO.TERMINATION_DATE, -999999999, -999999999,
       ocl.patient_pad(PAPO.PATIENT) PATIENT_ORDER
  from RXA_DES.PATIENT_POSITIONS PAPO
 where PAPO.CLINICAL_STUDY_ID = V_CLINICAL_STUDY_ID
   and PAPO.CLINICAL_STUDY_VERSION_ID = V_CLINICAL_STUDY_VERSION_ID
   and PAPO.HAS_DATA_FLAG = 'Y'
   and V_DATA_MODIFIED_FLAG = 'Y'
   /* patients with modified data and/or affected by lab changes */
   and (V_LAB_DEPENDENT_FLAG = 'Y'
       and ((PAPO.LAST_LAB_MODIFICATION_TS > PAPO.LAST_LAB_BATCH_TS) or
            (PAPO.DATA_MODIFIED_FLAG = 'Y')))
   and PAPO.OWNING_LOCATION = V_CURRENT_LOCATION
   and PAPO.FREEZE_FLAG = 'N'
   and V_TARGET_PATIENT = 0
UNION ALL
select PAPO.PATIENT_POSITION_ID, PAPO.CLINICAL_STUDY_ID, PAPO.REPORTED_SEX,
       PAPO.REPORTED_BIRTH_DATE, PAPO.PATIENT, PAPO.EARLY_TERMINATION_FLAG,
       PAPO.PATIENT_ENROLLMENT_DATE, PAPO.CLINICAL_SUBJECT_ID,
       PAPO.INCLUSION_EXCLUSION_DATE, PAPO.REPORTED_PATIENT_REFERENCE,
       PAPO.REPORTED_INITIALS, PAPO.REPORTED_DATE_LAST_PREGNANCY,
       PAPO.REPORTED_DEATH_DATE, PAPO.FIRST_SCREENING_DATE,
       PAPO.TERMINATION_DATE, -999999999, -999999999,
       ocl.patient_pad(PAPO.PATIENT) PATIENT_ORDER
  from RXA_DES.PATIENT_POSITIONS PAPO
 where PAPO.CLINICAL_STUDY_ID = V_CLINICAL_STUDY_ID
   and PAPO.CLINICAL_STUDY_VERSION_ID = V_CLINICAL_STUDY_VERSION_ID
   and PAPO.HAS_DATA_FLAG = 'Y'
   and PAPO.OWNING_LOCATION = V_CURRENT_LOCATION
   and PAPO.FREEZE_FLAG = 'N'
   /* single patient execution (DCAPI) */
   and (V_TARGET_PATIENT > 0 and PAPO.PATIENT_POSITION_ID = V_TARGET_PATIENT)
   /* Changing V_TARGET_PATIENT restriction to > 0 from <> 0 to implement RDC 3.2 branch for site */
order by 18;
```

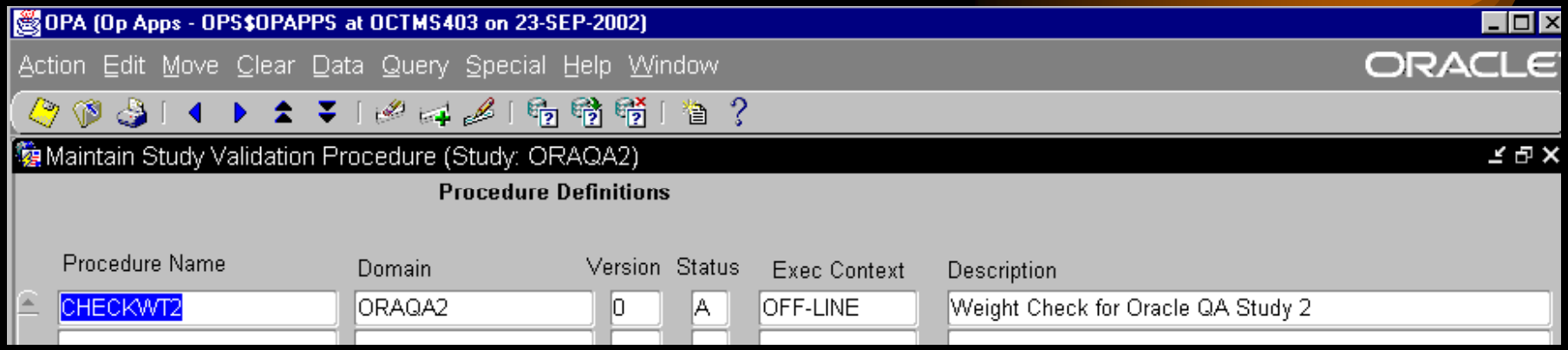
Analysis of the Package (2)

- The patients_cur cursor defines all patients for which this validation/derivation procedure will run.
  - When there is a problem with batch validation not seeing a specific patient, run this cursor query, substituting values for the v_ variables.
- For each patient_patient_position_id, the “D” cursor is opened. This is the cursor which was defined in the package header which finds the corresponding responses for a patient.
  - When there is a problem with a procedure that sees a patient, but not a specific response, run this cursor query, substituting the patient_position_id and the beginning and ending sequence number, which are also defined in the package header.

Analysis of the Package (3)

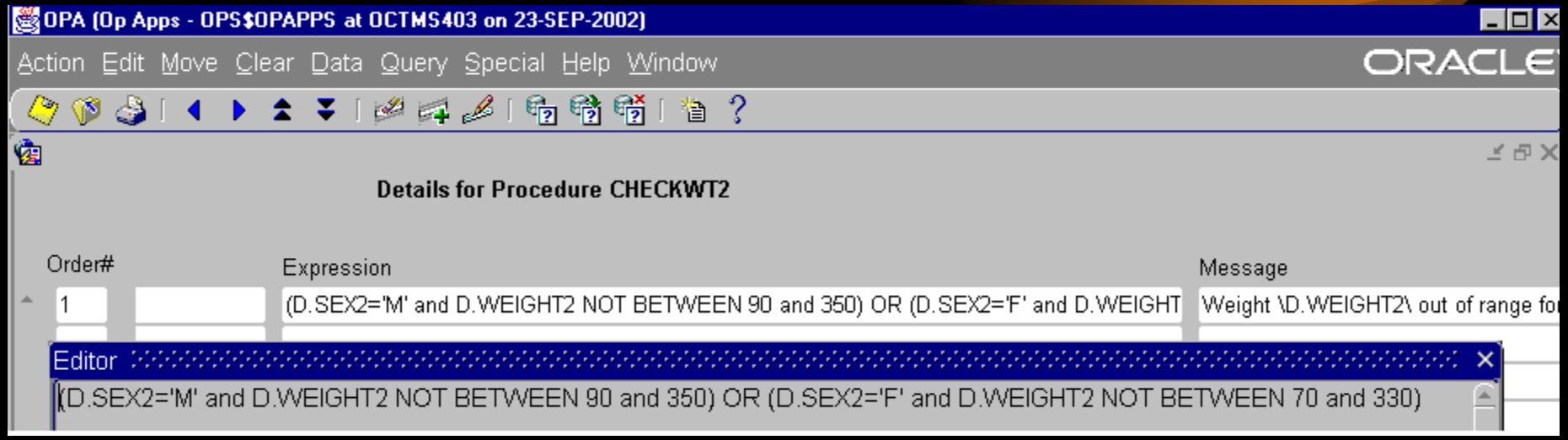
- For each response, the condition defined in the procedure details is evaluated. If the response fails the condition, then
  - The procedure rxcpdstd.check_discrepancy is called which determines if there is already an existing discrepancy.
  - If there is not an existing discrepancy, a new discrepancy is created via the insert_discrepancy private procedure, which in turn calls rxcpdstp.insert_discrepancy
  - Rows are added to the validation_reported_values table with the rxcpdstd.insert_vrv package.

A Basic Mapping to the OC Interface (1)



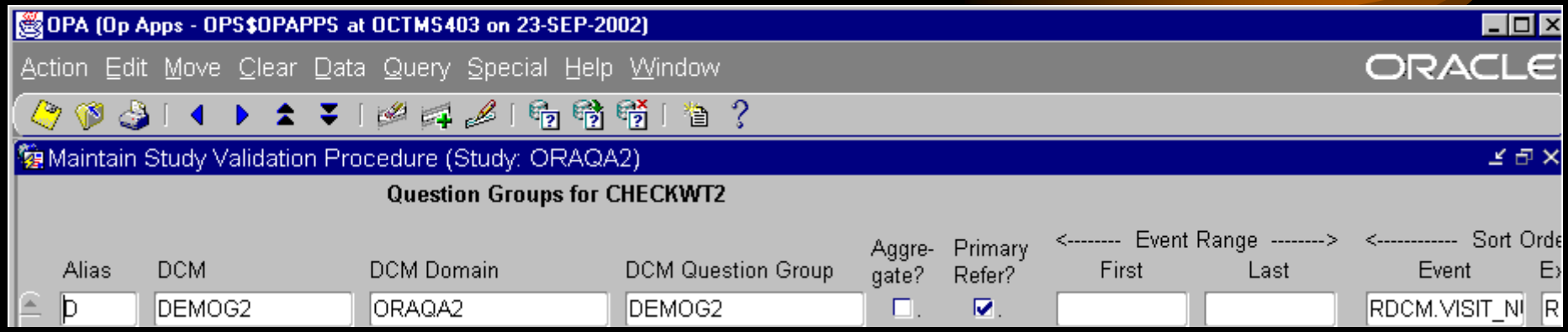
The Procedure Name and Version is stored in the RXC.PROCEDURES table and is used in the naming of the RXC_PD.RXCPD_<PROCEDURE_ID>_<PROCEDURE_VER_SN>

A Basic Mapping to the OC Interface (2)



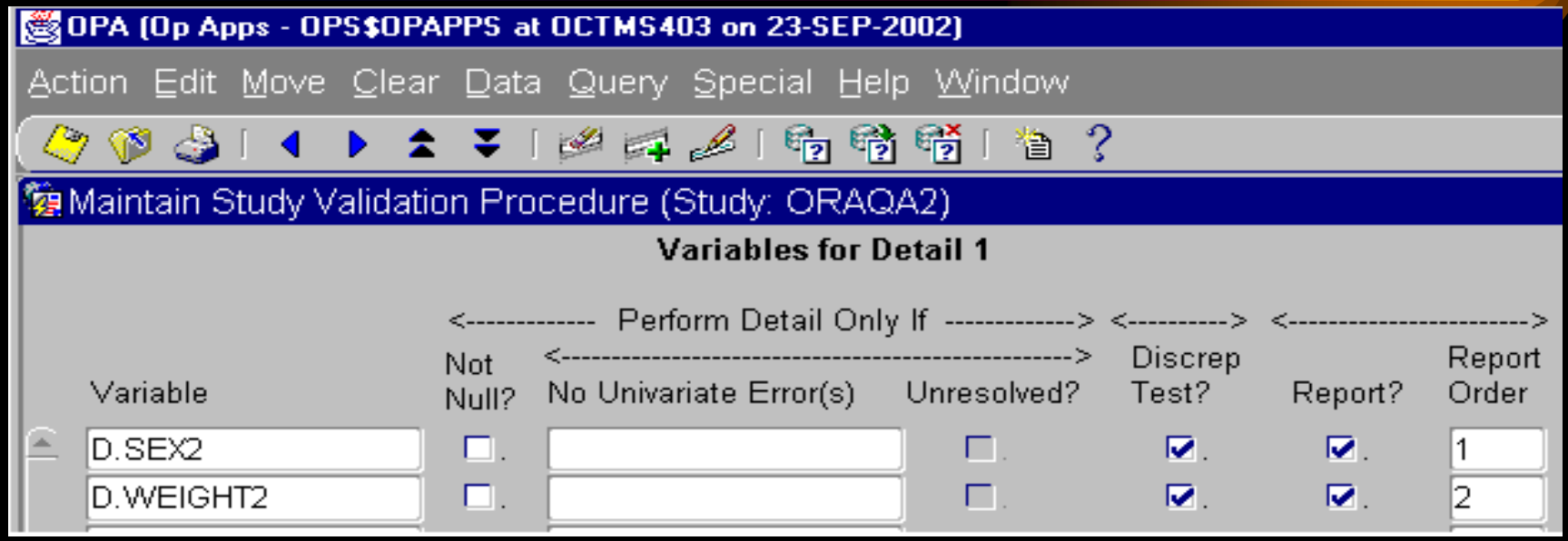
The details are stored in RXC.PROCEDURE_DETAILS and are the primary if condition for the nested cursor which evaluates responses.

A Basic Mapping to the OC Interface (3)



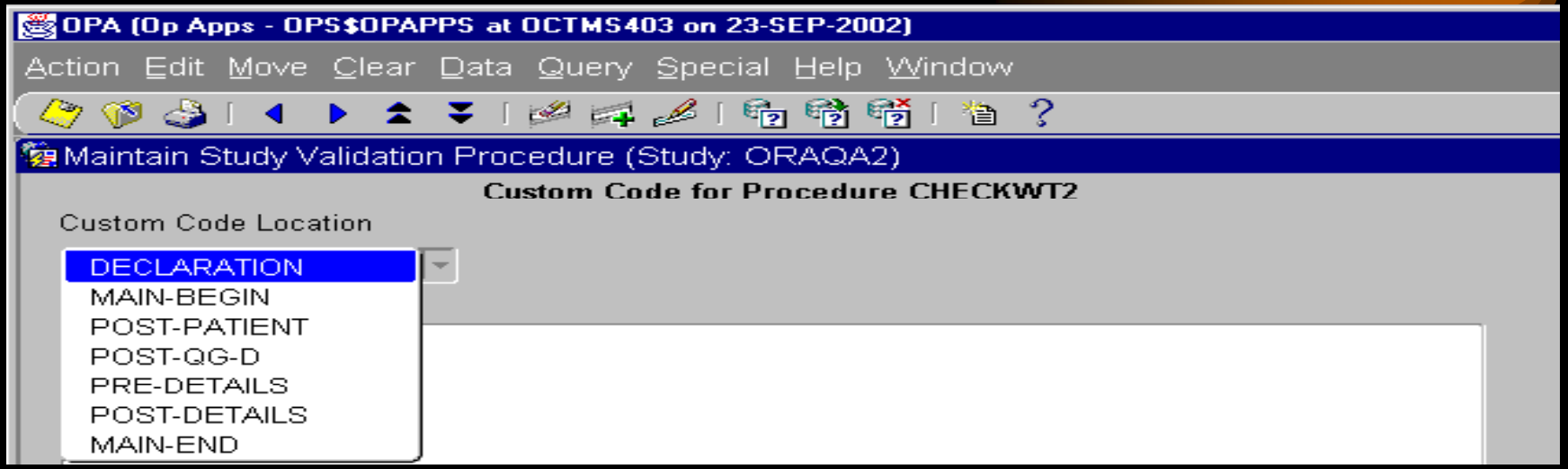
The Procedure Question group's alias is the associated to the nested cursor which retrieves responses as <alias>_cur. The QG ID and the DCM_ID are used as the criteria against received_dcms to join with responses. The Event Range are defined as the constants <alias>$begin_seqnum and <alias>$end_seqnum

A Basic Mapping to the OC Interface (4)



The procedure variables are built into the nested cursor named <alias>_cur which is in the package header for the responses. These names become the column aliases for the response.value_text

A Basic Mapping to the OC Interface (5)



The Procedure Details are inserted in the RXC_PD.RXCPD_<PROCEDURE_ID>_<PROCEDURE_VER_SN> between all points designated by /*<custom_code_location>*/

A Method for Expanding the Mapping

- From the existing queries and code for a simple procedure, changes can be individually. The resulting package's source can then be compared with this base package and the changes observed.
  - Adding additional details will add extra if conditions in the nested <QG_alias> cursor
  - Adding addition QGs will cause additional cursors to be created that recursively nest within each other. Be weary of Cartesian products in this scenario.

Useful Techniques Made Possible by Having the Package Source

- Interactive calls from the RXC_PD schema directly to MAIN procedure for debugging and timing purposes. The I_TIMER parameter can be set this way.
- A SQL script can be written around a call to this MAIN procedure which can turn on session level tracing and session level events.
- In a Development environment, this package can be customized and replaced, especially if trying to debug a problem with the procedure generation.

Developing a Function Library for Validation Procedures

- The package RXC.RXCPDSTD contains the common functions available to all validation/derivation procedures. A master list of pre-defined variables comes from this package.
- In the same manner, it is possible for organisations to make their own standard package, preferably in a custom schema. This package can contain all of the reusable functions, procedures, cursors and variables that can called by all validation/derivation procedures. After creating the package, execute grants must be made to rxclin_read, rxclin_mod, and rxc_pd. A public synonym can be optionally defined for ease of reference.

Q&A.

- Please come to Booth 19 and 20 with any additional questions, and also for:
  - Additional copies of this presentation
  - Free CD Cases
  - More Brochures, posters and cards
  - US Flag pin
- Presentation will also be on www.clinicalserver.com by end-of-week.