PROJECT

Claims Annual 2012-13 Report

DRAFT 1

Version 1.0

March 25, 2014

Version History

Date	Version #	Description	Author
03/25/2014	1.0a	Final Version	Vincent Taylor

TABLE OF CONTENTS

1 Introduction

This work request was submitted by Healthcare Claims Sales Department

1.1 Objectives

- Extract required fields from Data Ware House
- Create report as outlined in 1.3 Scope section.
- Delivery of files to requestor.

1.2 System Description

System/Application Name	Decision Support
Director	Bev
Requestor	Carrie

1.3 Scope

Create two reports. The first report, Claims 2012-13 will reflect Paid Year and Month, Employee Status, Plan Option, Number of Employees, Number of Members, Claim Paid In Network, Out of Network, Pharmacy Claims Paided and Total Claims paid. The second report will reflect year 2012-13 large claim information in excess of $50,000. Provide claim and enrollment data separated for any population grouping requiring separate rates.

1.4 Issues

None foreseen at this time.

1.5 Dependencies

N/A.

2 Execution

2.1 SAS Serve File

3 Environment

3.1 Hardware

N/A

3.2 Servers and Folders

affsanfe01
affsass00p01

3.3 Special Test Needs

N/A

4 Release Criteria

- System testing, independent testing and user acceptance testing must be carried out successfully
- Testing of all requirement enhancements, if any in the current release, must be completed

5 SAS Source Code

KC Claims 2012 2013 Annual Report	Completed
KC High Claim Amounts	Completed

KC Claims 2012-12 Annual Report

```
/*--------------*/
/* Timer Start */
/*--------------*/
data _null_;
     time = time();
     call symput('Start',time);
run;

options source2 symbolgen mautosource mprint sasautos=(m) mlogic mrecall;

%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments; * I Drive;
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;
%let DRIVE_G=\\affsass00p01;*SAS server;
%let OutDirYear = 2014; * HEDIS Technical Specifications for which the report
is running*;
%let HEDIS2 = HEDIS &OutDirYear\Stored Datasets; * Used for M;
PROC DATASETS LIBRARY=WORK MEMTYPE=DATA KILL NOLIST;
QUIT;
RUN;
options validvarname=v7;

/*----------------------------*/
/* Library \ file Definitions */
/*----------------------------*/
libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";
*libname HEDIS_QA "\\affsass00p01\sasdata\HEDIS\HEDIS 2014\Stored Datasets"; *
HEDIS Stored Datasets;
*libname DATAPATH "&DRIVE\&DATAPATH"; * Directory for Measures stored datasets;
*libname idss "&DRIVE\&IDSSPATH"; * output for idss;

/*-----------------------------*/
/* Global Variable Definitions */
/*-----------------------------*/
%let group  = "CORPORATION";    * Desired Group;
%let begdate  = '01JAN2012'd;
%let enddate =  '31DEC2013'd;
/*%let lob = 'HMO' 'NPP' 'POS'; */

/*-----------------------------------------------------------------*/
/* Read CLAIM_DETAIL_FACT Decision Support Warehouse SAS Dataset   */
/* create a subset of its contents using specific beg date and     */
/* group_dim group_key.                                            */
/*-----------------------------------------------------------------*/
/* Join all tables using KEYS                                      */
/* SET AS A - CREATE THE CLAIM_FACTS DATASET */

proc sql;
     create table claim_facts as
          select a.*,
                b.group_association_name,
                      b.group_employee_class,
                      b.group_employee_class_name
            from   star.claim_detail_fact a,
                   star.group_dim b
            where  b.group_name = &group                    and
                    a.group_key   = b.group_key              and
                   &begdate     <= a.payment_date <= &enddate and
                a.lob_code not in ('MED' ,'NPP')
     ;
quit;
/* Delete observations with paid_amt of zero commas */
proc sql;
     delete from claim_facts
            where paid_amt = 0;
quit;

/*----------------------------------------------------------------*/
/* Read MEMBER_MONTH_FACT Decision Support Warehouse SAS Dataset  */
/* to determine 0004 ACTIVE or 0005 COBRA status                  */
/*----------------------------------------------------------------*/
proc sql;
     create table member_month_fact as
         select a.member_id,
                   a.beg_date,
                a.group_key,
                      a.member_month_count,
                      a.subscriber_month_count,
                    b.group_employee_class,
                      b.group_employee_class_name,
                      b.group_association_name
         from   star.member_month_fact a,
                star.group_dim         b
            where  b.group_name = &group                  and
                   &begdate     <= a.beg_date <= &enddate and
                a.group_key  = b.group_key               and
                   b.group_employee_class in ('0004', '0005')
     ;

quit;

/*proc summary data=member_month_fact nway missing;*/
/*class group_association_name beg_date;*/
/*var Member_Month_Count Subscriber_Month_Count;*/
/*output out=mmfsum (drop=_type_ _freq)*/
/*sum=;*/
/*run;*/

/*-------------------------------------------------------------------*/
```

```
/* Read PHARMACY_DETAIL_FACT Decision Support Warehouse SAS Dataset  */
/* create a subset of its contents using specific fin period date,   */
/* also determine 0004 ACTIVE or 0005 COBRA status                   */
/*-------------------------------------------------------------------*/
proc sql;
     create table pharmacy_fact as
         select a.fin_period_date as payment_date,
                a.paid_amt as rxpaid_amt,
                      a.member_id,
                      a.beg_date,
                a.group_key,
                b.group_employee_class,
                      b.group_employee_class_name,
                      b.group_association_name
        from    star.pharmacy_detail_fact  a,
                   star.group_dim             b
           where   b.group_name = &group                        and
                &begdate     <= a.fin_period_date<= &enddate and
                a.group_key  = b.group_key                   and
                b.group_employee_class in ('0004', '0005')
      ;

quit;
/* Delete observations with paid_amt of zero commas */
proc sql;
     delete from pharmacy_fact
            where rxpaid_amt = 0;
quit;
/*-------------------------------------------------------------*/
/* Join with CLAIM AND CLAIM PROVIDER table to get In Network */
/*-------------------------------------------------------------*/

proc sql;
      create table claim_InNetWork as
            select a.member_id,
                   a.beg_date,
                a.payment_date,
                   a.paid_amt as inwpaid_amt,
                      a.lob_code,
                      a.group_key,
                      a.group_association_name,
                      a.group_employee_class_name,
                      a.group_employee_class,
                b.claim_prov_part_ind

         from   claim_facts                  a,
                   star.claim_provider_dim      b

          where  a.claim_prov_key       = b.claim_prov_key     and
                   B.claim_prov_part_ind  in ('Y')              and
                    a.group_employee_class in ('0004', '0005')

    ;
```

```
quit;

/* Get date for CLAIM_INETWORK_DTE  */

data claim_InNetWork_dte;

format paymonth mmddyy10. mth $8. mthyr $6.;

set claim_InNetWork;

paymonth = intnx('month', payment_date,0);
payyear  = year(paymonth);

/* format report monthe */
mth = compress(put(paymonth,mmddyy10.),'/');

/* Translate Paymonth Date */
if substr(mth,1,2) = '01'
   then mthyr = 'JAN-' || substr(mth,7,2);
if substr(mth,1,2) = '02'
   then mthyr = 'FEB-' || substr(mth,7,2);
if substr(mth,1,2) = '03'
   then mthyr = 'MAR-' || substr(mth,7,2);
if substr(mth,1,2) = '04'
   then mthyr = 'APR-' || substr(mth,7,2);
if substr(mth,1,2) = '05'
   then mthyr = 'MAY-' || substr(mth,7,2);
if substr(mth,1,2) = '06'
   then mthyr = 'JUN-' || substr(mth,7,2);
if substr(mth,1,2) = '07'
   then mthyr = 'JUL-' || substr(mth,7,2);
if substr(mth,1,2) = '08'
   then mthyr = 'AUG-' || substr(mth,7,2);
if substr(mth,1,2) = '09'
   then mthyr = 'SEP-' || substr(mth,7,2);
if substr(mth,1,2) = '10'
   then mthyr = 'OCT-' || substr(mth,7,2);
if substr(mth,1,2) = '11'
   then mthyr = 'NOV-' || substr(mth,7,2);
if substr(mth,1,2) = '12'
   then mthyr = 'DEC-' || substr(mth,7,2);

run;
/*-------------------------------------------------------------*/
/* Join with CLAIM AND CLAIM PROVIDER table to get Out Network */
/*-------------------------------------------------------------*/

proc sql;
      create table claim_OutNetWork as
            select a.member_id,
                   a.beg_date,
                a.payment_date,
                   a.paid_amt as onwpaid_amt,
                      a.lob_code,
                      a.group_key,
                      a.group_association_name,
                      a.group_employee_class_name,
                      a.group_employee_class,
                b.claim_prov_part_ind

         from   claim_facts                  a,
                   star.claim_provider_dim      b

          where  a.claim_prov_key       = b.claim_prov_key     and
                   b.claim_prov_part_ind  in ('N')               and
                    a.group_employee_class in ('0004', '0005')

    ;

quit;

/* Get date for CLAIM_OUTNETWORK_DTE  */
data claim_OutNetWork_dte;

format paymonth mmddyy10. mth $8. mthyr $6.;

set claim_OutNetWork;

paymonth = intnx('month', payment_date,0);
payyear  = year(paymonth);

/* format report monthe */
mth = compress(put(paymonth,mmddyy10.),'/');

/* Translate Paymonth Date */
if substr(mth,1,2) = '01'
   then mthyr = 'JAN-' || substr(mth,7,2);
if substr(mth,1,2) = '02'
   then mthyr = 'FEB-' || substr(mth,7,2);
if substr(mth,1,2) = '03'
   then mthyr = 'MAR-' || substr(mth,7,2);
if substr(mth,1,2) = '04'
   then mthyr = 'APR-' || substr(mth,7,2);
if substr(mth,1,2) = '05'
   then mthyr = 'MAY-' || substr(mth,7,2);
if substr(mth,1,2) = '06'
   then mthyr = 'JUN-' || substr(mth,7,2);
if substr(mth,1,2) = '07'
   then mthyr = 'JUL-' || substr(mth,7,2);
if substr(mth,1,2) = '08'
   then mthyr = 'AUG-' || substr(mth,7,2);
if substr(mth,1,2) = '09'
   then mthyr = 'SEP-' || substr(mth,7,2);
if substr(mth,1,2) = '10'
   then mthyr = 'OCT-' || substr(mth,7,2);
if substr(mth,1,2) = '11'
   then mthyr = 'NOV-' || substr(mth,7,2);
if substr(mth,1,2) = '12'
```

```
   then mthyr = 'DEC-' || substr(mth,7,2);

run;

/* Get date for MEMBER_MONTH_FACT_SUMMARY  */

data member_month_fact_dte ;

format paymonth mmddyy10. mth $8. mthyr $6. ;

set member_month_fact;

paymonth = intnx('month', beg_date,0);
payyear  = year(paymonth);

/* format report month */
mth = compress(put(paymonth,mmddyy10.),'/');

/* Translate Paymonth Date */
if substr(mth,1,2) = '01'
   then mthyr = 'JAN-' || substr(mth,7,2);
if substr(mth,1,2) = '02'
   then mthyr = 'FEB-' || substr(mth,7,2);
if substr(mth,1,2) = '03'
   then mthyr = 'MAR-' || substr(mth,7,2);
if substr(mth,1,2) = '04'
   then mthyr = 'APR-' || substr(mth,7,2);
if substr(mth,1,2) = '05'
   then mthyr = 'MAY-' || substr(mth,7,2);
if substr(mth,1,2) = '06'
   then mthyr = 'JUN-' || substr(mth,7,2);
if substr(mth,1,2) = '07'
   then mthyr = 'JUL-' || substr(mth,7,2);
if substr(mth,1,2) = '08'
   then mthyr = 'AUG-' || substr(mth,7,2);
if substr(mth,1,2) = '09'
   then mthyr = 'SEP-' || substr(mth,7,2);
if substr(mth,1,2) = '10'
   then mthyr = 'OCT-' || substr(mth,7,2);
if substr(mth,1,2) = '11'
   then mthyr = 'NOV-' || substr(mth,7,2);
if substr(mth,1,2) = '12'
   then mthyr = 'DEC-' || substr(mth,7,2);

run;

/* Get date for PHARMACY_FACT_DTE  */

data pharmacy_fact_dte ;

format paymonth mmddyy10. mth $8. mthyr $6. ;

set pharmacy_fact;
```

```
paymonth = intnx('month', payment_date,0);
payyear  = year(paymonth);


/* format report monthe */
mth = compress(put(paymonth,mmddyy10.),'/');

/* Translate Paymonth Date */
if substr(mth,1,2) = '01'
   then mthyr = 'JAN-' || substr(mth,7,2);
if substr(mth,1,2) = '02'
   then mthyr = 'FEB-' || substr(mth,7,2);
if substr(mth,1,2) = '03'
   then mthyr = 'MAR-' || substr(mth,7,2);
if substr(mth,1,2) = '04'
   then mthyr = 'APR-' || substr(mth,7,2);
if substr(mth,1,2) = '05'
   then mthyr = 'MAY-' || substr(mth,7,2);
if substr(mth,1,2) = '06'
   then mthyr = 'JUN-' || substr(mth,7,2);
if substr(mth,1,2) = '07'
   then mthyr = 'JUL-' || substr(mth,7,2);
if substr(mth,1,2) = '08'
   then mthyr = 'AUG-' || substr(mth,7,2);
if substr(mth,1,2) = '09'
   then mthyr = 'SEP-' || substr(mth,7,2);
if substr(mth,1,2) = '10'
   then mthyr = 'OCT-' || substr(mth,7,2);
if substr(mth,1,2) = '11'
   then mthyr = 'NOV-' || substr(mth,7,2);
if substr(mth,1,2) = '12'
   then mthyr = 'DEC-' || substr(mth,7,2);

run;

/* Perform Claims InNetWork Summary */

proc summary data=claim_InNetWork_dte nway missing;
    class group_association_name paymonth member_id;
      *id member_id;
      var inwpaid_amt;
      output out=claim_InNetWork_summary(drop=_type_ _freq_ _stat_)
      sum=;
run;

proc sort data=claim_InNetWork_summary;
 by paymonth group_association_name member_id;
run;

/* Perform Claims OutNetWork Summary */

proc summary data=claim_OutNetWork_dte nway missing;
    class group_association_name paymonth member_id;
      *id member_id;
      var onwpaid_amt;
      output out=claim_OutNetWork_summary(drop=_type_ _freq_ _stat_)
      sum=;
run;

proc sort data=claim_OutNetWork_summary;
 by paymonth group_association_name member_id;
run;

/* Perform Pharmacy Summary */

proc summary data=pharmacy_fact_dte nway missing;
    class group_association_name paymonth member_id;
      *id member_id;
      var rxpaid_amt;
      output out=pharmacy_fact_summary(drop=_type_ _freq_ _stat_)
      sum=;
run;

proc sort data=pharmacy_fact_summary;
 by paymonth group_association_name member_id;
run;

/* Perform Member_Month_Fact Summary */

proc summary data=member_month_fact_dte nway missing;
    class group_association_name  paymonth;
      *id member_month_count subscriber_month_count;
      var member_month_count subscriber_month_count;
      output out=member_month_fact_summary(drop=_type_ _freq_ _stat_)
      sum=;
run;
proc sort data=member_month_fact_summary;
by paymonth group_association_name;

/*---------------------------*/
/* Merge CLAIM and PHARMACY  */
/*---------------------------*/
data claims_pharm;
  merge claim_InNetWork_summary claim_OutNetWork_summary pharmacy_fact_summary;
    by paymonth group_association_name member_id;
 run;

/*-----------------------------*/
/* Sort CLAIMS_PHARM Dataset   */
/*-----------------------------*/
proc sort data=claims_pharm;
by paymonth member_id;
run;

/* create claim pharm and member month fact table */
proc sql;
     create table claims_pharm_mfs as
         select distinct
```

```
                   a.*,
                   b.lob_code,
                      b.group_employee_class,
                c.member_month_count,
                c.subscriber_month_count
         from   claims_pharm                a,
                   claim_InNetWork             b,
                   member_month_fact_summary   c

            where  a.paymonth = b.beg_date      and
                a.paymonth = c.paymonth
     ;

quit;
/*---------------------------------*/
/* Sort CLAIMS_PHARM_mfs Dataset   */
/*--------------------------------*/
proc sort data=claims_pharm_mfs nodupkey;
by paymonth member_id;
run;
/*------------------------------*/
/* Process merged claims data   */
/*------------------------------*/
data merge_data grand_data;

format employee_status $12. plan_option $11. mthyr $6. inwpaid_amt comma13.2
onwpaid_amt comma13.2
       rxpaid_amt comma13.2 tot_claims_paid comma13.2 grand_inwpaid_amt
comma13.2
       grand_onwpaid_amt comma13.2 grand_rxpaid_amt comma13.2
grand_tot_claims_paid comma13.2;

retain grand_inwpaid_amt grand_onwpaid_amt grand_rxpaid_amt
grand_tot_claims_paid 0;

/* Translate Employee Status */

employee_status = 'Active/COBRA';

plan_option = 'Network HMO';

/* format report monthe */
mth = compress(put(paymonth,mmddyy10.),'/');

/* Translate Paymonth Date */
if substr(mth,1,2) = '01'
   then mthyr = 'JAN-' || substr(mth,7,2);
if substr(mth,1,2) = '02'
   then mthyr = 'FEB-' || substr(mth,7,2);
if substr(mth,1,2) = '03'
   then mthyr = 'MAR-' || substr(mth,7,2);
if substr(mth,1,2) = '04'
   then mthyr = 'APR-' || substr(mth,7,2);
if substr(mth,1,2) = '05'
```

```
   then mthyr = 'MAY-' || substr(mth,7,2);
if substr(mth,1,2) = '06'
   then mthyr = 'JUN-' || substr(mth,7,2);
if substr(mth,1,2) = '07'
   then mthyr = 'JUL-' || substr(mth,7,2);
if substr(mth,1,2) = '08'
   then mthyr = 'AUG-' || substr(mth,7,2);
if substr(mth,1,2) = '09'
   then mthyr = 'SEP-' || substr(mth,7,2);
if substr(mth,1,2) = '10'
   then mthyr = 'OCT-' || substr(mth,7,2);
if substr(mth,1,2) = '11'
   then mthyr = 'NOV-' || substr(mth,7,2);
if substr(mth,1,2) = '12'
   then mthyr = 'DEC-' || substr(mth,7,2);

set claims_pharm_mfs;
  by paymonth member_id;

if lob_code ^= 'HMO' then delete;

if ^last.paymonth then
   do;

    if inwpaid_amt =. then inwpaid_amt = 0;
    if onwpaid_amt =. then onwpaid_amt = 0;
    if rxpaid_amt  =. then rxpaid_amt = 0;

    /* total claims paid for all members during the month */
    tot_claims_paid = sum(of inwpaid_amt onwpaid_amt rxpaid_amt);

     /* calculate grand_totals */
    grand_inwpaid_amt+inwpaid_amt;
    grand_onwpaid_amt+onwpaid_amt;
    grand_rxpaid_amt+rxpaid_amt;

    output merge_data;

   end;

if last.paymonth then
   do;
      /* calculate grand_totals */
      if inwpaid_amt =. then inwpaid_amt = 0;
      if onwpaid_amt =. then onwpaid_amt = 0;
      if rxpaid_amt  =. then rxpaid_amt = 0;
      grand_inwpaid_amt+inwpaid_amt;
      grand_onwpaid_amt+onwpaid_amt;
      grand_rxpaid_amt+rxpaid_amt;
      grand_tot_claims_paid = sum(of grand_inwpaid_amt grand_onwpaid_amt
grand_rxpaid_amt);

       output grand_data;
       grand_inwpaid_amt = 0;
      grand_onwpaid_amt = 0;
      grand_rxpaid_amt = 0;
      grand_tot_claims_paid = 0;

   end;
run;
/*-----------------------------*/
/* Sort MERGE_DATA Dataset     */
/*-----------------------------*/
proc sort data=grand_data;
by paymonth employee_status plan_option subscriber_month_count
member_month_count grand_inwpaid_amt
   grand_onwpaid_amt grand_rxpaid_amt grand_tot_claims_paid;
run;

/*--------------------------*/
/* Prepare report Dataset   */
/*--------------------------*/
data merge_final (keep=MTHYR
                       EMPLOYEE_STATUS
                       PLAN_OPTION
                       NUMBER_OF_EMPLOYEES
                       NUMBER_OF_MEMBERS
                       CLAIMS_PAID_IN_NETWORK_MEDICAL
                       CLAIMS_PAID_OUT_NETWORK_MEDICAL
                       PHARMACY_PAID_CLAIMS
                       TOTAL_CLAIMS_PAID )
                               ;

    retain             MTHYR
                       EMPLOYEE_STATUS
                       PLAN_OPTION
                       NUMBER_OF_EMPLOYEES
                       NUMBER_OF_MEMBERS
                       CLAIMS_PAID_IN_NETWORK_MEDICAL
                       CLAIMS_PAID_OUT_NETWORK_MEDICAL
                       PHARMACY_PAID_CLAIMS
                       TOT_CLAIMS_PAID
                               ;
   format CLAIMS_PAID_IN_NETWORK_MEDICAL   dollar13.2
          CLAIMS_PAID_OUT_NETWORK_MEDICAL  dollar13.2
          PHARMACY_PAID_CLAIMS             dollar13.2
          TOTAL_CLAIMS_PAID                dollar13.2;

   set grand_data;

   CLAIMS_PAID_IN_NETWORK_MEDICAL = grand_inwpaid_amt;
   CLAIMS_PAID_OUT_NETWORK_MEDICAL = grand_onwpaid_amt;
   PHARMACY_PAID_CLAIMS = grand_rxpaid_amt;
   TOTAL_CLAIMS_PAID = grand_tot_claims_paid;
   rename subscriber_month_count=NUMBER_OF_EMPLOYEES;
   rename member_month_count=NUMBER_OF_MEMBERS;
```

```
run;

filename OutClaim "&DRIVE_I.\DecisionSupport\vtaylor\SAS\OutputFiles\Summary of
Claims 2012 2013 Annual Report &SYSDATE9..xlsx";
/*-------------------------------------*/
/* Write CLAIM Dataset to excel file  */
/*-------------------------------------*/
proc export data=merge_final
     outfile=OutClaim
      dbms=EXCEL2002 replace;
run;


*-----------*
* Timer End *
*-----------*;
data _null_;
      format Start 8.;
      Time = Time();
      Start = symget('Start');
      Duration = Time - Start;
      put '****** Start Time= ******' Start  time12.5;
      put '****** End   Time= ******' Time   time12.5;
      put '****** Duration  = ******' Duration time12.2;
run;

quit;
```

KC High Claim Amounts

```
data _null_;
      time = time();
      call symput('Start',time);
run;
options symbolgen missing=0;

%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments_Network_Health; * I
Drive;
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;
%let DRIVE_G=\\affsass00p01;*server;
options validvarname=v7;
libname star clear;
libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";*for reporting after audits
are approved;
*libname star "&DRIVE_G\sasdata\DEV\SAS Warehouse\Star";*for audits;

DATA FMTDX (KEEP=FMTNAME TYPE START LABEL HLO);        /* DXCODE */
 length label $10;
  SET STAR.DIAGNOSIS_DIM END=LAST;
  FMTNAME='DX_CODE';
  TYPE='C';
  START=DIAGNOSIS_KEY;
  LABEL=DIAGNOSIS_CODE;
  HLO=' ';
  OUTPUT;
  IF LAST THEN DO;
   HLO='O';
   LABEL='None Found';
   OUTPUT;
  END;
  RUN;
  PROC FORMAT LIBRARY=work CNTLIN=FMTDX;
  RUN;
  PROC DATASETS LIBRARY=WORK;
  DELETE FMTDX;
  QUIT;
  RUN;

  /******************************DIAGNOSIS
DESCRIPTION*************************************************/
  DATA FMTDXDESC (KEEP=FMTNAME TYPE START LABEL HLO);        /* DXDESC */
 length label $25;
  SET STAR.DIAGNOSIS_DIM END=LAST;
  FMTNAME='DX_DESCRIPTION';
  TYPE='C';
  START=DIAGNOSIS_KEY;
  LABEL=DIAGNOSIS_DESC;
  HLO=' ';
  OUTPUT;
  IF LAST THEN DO;
   HLO='O';
   LABEL='None Found';
   OUTPUT;
  END;
  RUN;
  PROC FORMAT LIBRARY=work CNTLIN=FMTDXDESC;
  RUN;
  PROC DATASETS LIBRARY=WORK;
  DELETE FMTDXDESC;
  QUIT;
  RUN;

*%let Assoc='SARGENTO FOODS INC';
%let Assoc='CORPORATION';

proc datasets library=work memtype=data kill;
quit;
run;

data group_dim;
set star.group_dim;
if group_association_name = 'CORPORATION' then do;
      if group_employee_class in('0007', 'E-RTR') then group_association_name =
'CORPORATION-Retirees';
end;
```

```
if group_association_name in(&Assoc);
if group_affiliation_name in('SENIOR PLUS' 'LARGE GROUP') then Size =
group_affiliation_name;
     else if group_affiliation_name = 'MID-SIZE GROUP' then Size = 'LARGE
GROUP';
          else do;
               size = 'SMALL GROUP';
          end;

          *if Group_Affiliation_Name in(&Affil);

          *if Group_Affiliation_Name='FOX CITIES CHAMBER';
run;

********* Create DATE Macro Variables ************************;
data _null_;


     logdate=trim(left(compress(put(today(),date9.),'-/')));
     call symput('logdate',logdate);

     begdate=intnx('month', today(), -13);       * Default=-15 (Rolling M12 90
days out);
     begdate='01JAN2012'D;
     call symput("BEGDOS", put(begdate, date9.));
     enddate=intnx('month', today(), -0)-1;      * Default = -3 (90 day lag);
     enddate='31DEC2013'D;
     call symput("ENDDOS", put(enddate, date9.));
     CurrBOM = intnx('month',today(),-1);        * Default = -1 (latest month
in warehouse);
     call symput("BEGDAT", put(CurrBOM, date9.));
     CurrEOM = intnx('month',today(),0)-1;
     call symput("ENDDAT", put(CurrEOM, date9.));

run;

%put &BEGDOS &ENDDOS &logdate &Begdat &EndDat;

data product_dim;
set star.product_dim (where=(lob_code not in('MED' 'NPP')));/*add MED PER BEV
VER VOORT 12/18/2007*/
RUN;

proc sql;
     create table medclm   as
     select     q.payment_date,
                q.paid_amt as medpaid_amt,
                q.product_key,
                q.group_key,
                q.member_id,
                q.lob_code,
                q.first_dx_key,
                c.age,
            c.sex,
                i.benefit_plan_code,
                i.contract_type,
                i.contract_desc,
                p.group_association_name,
                p.group_name,
                q.data_source,
                q.document_number
     from  work.product_dim i,
                star.age_sex_dim c,
                work.group_dim        p,
                star.claim_detail_fact     q
     where "&BEGDOS"d <=q.payment_date<="&ENDDOS"d                     and
                p.group_key                      =q.group_key
     and
                c.age_sex_key           = q.age_sex_key              and
                i.product_key=q.product_key;
                quit;


data medclm;
set medclm;
paymonth = intnx('month', payment_date, 0); format paymonth mmddyy10.;
PAYYEAR=YEAR(PAYMONTH);
  first_dx_code=put(first_dx_key,$DX_CODE.);
  first_dx_desc=put(first_dx_key,$DX_DESCRIPTION.);

run;

/*summarize by payyear group member and primary diagnosis to get the most
expensive diagnosis*/
proc summary data=medclm nway missing;
class PAYYEAR Group_Association_Name member_id first_dx_code;
id first_dx_desc;
var medpaid_amt;
output out=first_dx_sum
sum=;
run;
proc sort data=first_dx_sum;
by PAYYEAR Group_Association_Name member_id descending medpaid_amt;
run;

/* this keep the most expensive diagnosis paid during the year*/
data first_dx_sum;
set first_dx_sum;
by PAYYEAR Group_Association_Name member_id;
if first.member_id;
keep PAYYEAR Group_Association_Name member_id first_dx_code first_dx_desc;
run;

/*get the pharmacy claims by payment date*/

proc sql;
     create table rxclm    as
     select /*  q.beg_date as payment_date,*/
                q.Fin_Period_Date as payment_date,
                q.paid_amt as rxpaid_amt,
                q.product_key,
                q.group_key,
                q.member_id,
                q.lob_code,
                c.age,
            c.sex,
                i.benefit_plan_code,
                i.contract_type,
                i.contract_desc,
                p.group_association_name,
                p.group_name,
                q.data_source,
                q.document_number
     from  work.product_dim i,
                star.age_sex_dim c,
                work.group_dim       p,
                star.pharmacy_detail_fact  q
     where "&BEGDOS"d <=q.Fin_Period_Date<="&ENDDOS"d                  and
                p.group_key                      =q.group_key
          and
                c.age_sex_key           = q.age_sex_key
and
                i.product_key=q.product_key;
                quit;

/*changing the paid date to the begining of the month*/
data rxclm;
set rxclm;
paymonth = intnx('month', payment_date, 0); format paymonth mmddyy10.;
PAYYEAR=YEAR(PAYMONTH);
run;
/*data rxclm531;*/
/*set rxclm;*/
/*run;*/

/*get the maximum age for each year a member has a claim*/
data age;
set rxclm medclm;
keep PAYYEAR group_association_name member_id age ;
run;
proc sort data=age;
by PAYYEAR group_association_name member_id descending age;
run;

data age;
set age;
by PAYYEAR group_association_name member_id descending age;
if first.member_id;
run;
```

```
/*summarizing by member to get members that had total paid amounts in excess of
$150,000 within the review period*/

proc summary data=medclm nway missing;
class PAYYEAR group_association_name member_id;
id sex;
var medpaid_amt;
output out=memmedclmsum (drop=_type_ _freq_)
sum=;
run;
proc summary data=rxclm nway missing;
class PAYYEAR group_association_name member_id;
id sex;
var rxpaid_amt;
output out=memrxclmsum (drop=_type_ _freq_)
sum=;
run;

data members_claim50;
merge memmedclmsum memrxclmsum;
by PAYYEAR group_association_name member_id;
if medpaid_amt=. then medpaid_amt=0;
if rxpaid_amt=. then rxpaid_amt=0;
tot_paid_amt=sum(of medpaid_amt rxpaid_amt);
if tot_paid_amt ge 50000;*important;
KEEP PAYYEAR MEMBER_ID sex GROUP_ASSOCIATION_NAME medpaid_amt rxpaid_amt
tot_paid_amt;
run;

proc sql;
      create table mmf as
      select distinct
                 a.*,
                 r.disenroll_date,
                 O.FAMILY_ID,
                 o.member_name
      from  work.group_dim             p,
                 STAR.MEMBER_DIM            O,
                 work.members_claim50  a,
                 star.member_month_fact     r
      where      a.member_id=r.member_id    and
                 a.member_id=O.member_id    and
                 r.group_key=p.group_key;
                 quit;
      PROC SORT DATA=mmf;
      BY PAYYEAR GROUP_ASSOCIATION_NAME MEMBER_ID DESCENDING DISENROLL_DATE;
      RUN;

data mmf;
set mmf;
BY PAYYEAR GROUP_ASSOCIATION_NAME MEMBER_ID DESCENDING DISENROLL_DATE;
IF FIRST.MEMBER_ID;
MEMBER_STATUS='Termed';
if disenroll_date ge today() then MEMBER_STATUS='Active';
```

```
RUN;

DATA MMF;
MERGE MMF (IN=A) AGE (IN=B) first_dx_sum (IN=C);
BY PAYYEAR GROUP_ASSOCIATION_NAME MEMBER_ID;
IF A;
RUN;

/*THE FOLLOWING SCRAMBLES THE MEMBER_ID*/
      DATA MMF2;
      SET MMF;
      BY PAYYEAR GROUP_ASSOCIATION_NAME MEMBER_ID;
      LENGTH=LENGTH(MEMBER_ID);/*CALCULATE THE LENGTH OF MEMBER_ID*/
      IF MEMBER_ID ^= FAMILY_ID AND MEMBER_ID ^= '0' THEN
      OLD_member_id=SUBSTR(MEMBER_ID,1,LENGTH-2)||'*'||SUBSTR(MEMBER_ID,LENGTH-
1,2);/*ADDING ASTERISK*/
           ELSE OLD_MEMBER_ID=MEMBER_ID;

PERSNO=upcase(scan(old_member_ID,2,"*"));/*LAST 2 POSITIONS OF THE MEMBER_ID*/
EMPLOYEE_ID=TRIM(INPUT((FAMILY_ID*9),$12.));/*SCRAMBLING THE FAMILY ID*/
MEMBNO=TRIM(COMPRESS(EMPLOYEE_ID||'*'||PERSNO));/* CREATING A SCRABLED
MEMBER_ID*/
*DROP LENGTH FAMILY_ID OLD_member_id EMPLOYEE_ID PERSNO MEMBER_ID medpaid_amt
rxpaid_amt;
KEEP PAYYEAR Group_Association_Name MEMBER_ID MEMBNO ;* member_id family_id
member_name;
RUN;

DATA MMF3;
retain  GROUP_ASSOCIATION_NAME  MEMBNO MEMBER_STATUS PAYYEAR  AGE SEX
first_dx_code first_dx_desc
medpaid_amt rxpaid_amt tot_paid_amt;
MERGE MMF (IN=A) MMF2 (IN=B);
BY PAYYEAR GROUP_ASSOCIATION_NAME MEMBER_ID;
IF A;
RENAME MEMBNO=CLAIMANT_NUMBER;
RENAME SEX=GENDER;
RENAME medpaid_amt=MEDICAL_PAID_AMT;
RENAME rxpaid_amt=RX_PAID_AMT;
RENAME tot_paid_amt=TOTAL_PAID_AMT;
KEEP  GROUP_ASSOCIATION_NAME  MEMBNO MEMBER_STATUS PAYYEAR  AGE SEX
first_dx_code first_dx_desc
medpaid_amt rxpaid_amt tot_paid_amt;
RUN;
proc sort data=mmf3;
by GROUP_ASSOCIATION_NAME PAYYEAR descending TOTAL_PAID_AMT;
run;

/* this is for the summary by group_association_name members with total claims
in excess of $150,000 */
DATA EXCEL2;
SET mmf3;
daterange=put("&BEGDOS"D,MMDDYY10.)||' to '||PUT("&ENDDOS"D,MMDDYY10.);
paidthru=PUT("&EndDat"D,MMDDYY10.);
```

```
rundate=today();
format rundate mmddyy10.;
RUN;

PROC EXPORT DATA= work.excel2
            OUTFILE=
"&DRIVE_I\DecisionSupport\Carlos\EXCEL\OUTPUT\GROUP_SALES\HighDollar_SUMMARY_TE
MPLATE.xls"
            DBMS=EXCEL2002 REPLACE;
                SHEET="HIGHDOLLAR_DATA";
RUN;

/*
PROC DATASETS LIBRARY=WORK MEMTYPE=DATA KILL;
QUIT;
RUN;
*/

*************
* Timer End *
*************;
data _null_;
      format Start 8.;
      Time = Time();
      Start = symget('Start');
      Duration = Time - Start;
      put '****** Start Time= ******' Start  time12.5;
      put '****** End   Time= ******' Time   time12.5;
      put '****** Duration  = ******' Duration time12.2;
run;

quit;
```