PROJECT

Terminating Providers

DRAFT 1

Version 1.0

September 18, 2014

Version History

Date	Version #	Description	Author
09/18/2014	1.0a	Final Version	Vincent Taylor

TABLE OF CONTENTS

Terminating Providers

1   Introduction

This work request was initiated by DecisionSuppport

1.1   Objectives

- Further automate the terminating provider process handled by Enrollment Services
- To shorten the time it takes DecisionSupport to complete Terminating Provider request

1.2   System Description

System/Application Name	Decision Support
Director	Bev
Requestor	Vincent Taylor

1.3   Scope

Inserted macro functionality to convert the execution of the programs used in the Terminating Provider process so that multiple providers can be processed at a time..

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

Mental Health Provider Rev7 Sep17	Modifications completed
Non Mental Health Rev9 Sep16	Modifications completed
PCP Term Ver7 SEP16	Modifications complete

Mental Health Provider Rev7 SEP17

```
***************
* Timer Start *
***************;
data _null_ ;
    time = time();
    call symput('Start',time);
run;
ods listing;
********* Assign Libraries ************************************;
 *** WRITE TO TERMNATING PROVIDER FOLDER ***;
%let driveserv = \\affsass00p01;
%let driveI = \\affsanfe01\AFF_Departments\Departments;
%let drivey = \\affsanfe01\Share_Team;
%let SAS_PGM_LOC_PROD= &driveserv\sasdata\RoutineReports\Monthly;
filename m "&DRIVEI\DecisionSupport\HEDIS 2013\SAS Programs\Macros";
options source2 symbolgen mautosource mprint sasautos=(M) mlogic mrecall;  *
Sets up the macro library;

libname PERMSAS clear;
libname PERMSAS "&driveI\DecisionSupport\RoutineReports\On Demand\
Terminating_Provider_Lists\permsas";
libname prodsas "&SAS_PGM_LOC_PROD\Membership\Membership_Redesign\DataLibrary";

 *** READ FROM FILE WITH MOST RECENT MEMBER AND SUBSCRIBER ADDRESS ***;
Libname SUMMARY clear;
libname SUMMARY "&driveI\DecisionSupport\RoutineReports\Monthly\
Membership\current mbrs";

 *** READ FROM THE DATA WAREHOUSE ***;
libname star clear;
/*libname STAR "C:\SAS Warehouse\Star";*/
libname star '\\affsass00p01\sasdata\SAS Warehouse\Star';

******************** Parameters & Constants ****************************;
*** ENTER BEGIN / END DATES OF SERVICE TO REFLECT 12 MONTH TIME-FRAME ***;
**********************************************************************;
********** verify and change these before each run of program **********;
**********************************************************************;
ods listing;
options symbolgen;
   %macro fdate;    /* 092014 */
       %global FDATE1 TERM_DATE MM PREV_YEAR CUR_YEAR CUR_CC;
```

Terminating Providers

```sas
data _null_;

   %let dd1 = 31DEC;
   %let dd2 = 31JAN;
   %let dd3 = 28FEB;
   %let dd4 = 31MAR;
   %let dd5 = 30APR;
   %let dd6 = 31MAY;
   %let dd7 = 30JUN;
   %let dd8 = 31JUL;
   %let dd9 = 31AUG;
  %let dd10 = 30SEP;
  %let dd11 = 31OCT;
  %let dd12 = 29NOV;
  %let diff = 1;
  %let operator=-;

 %let FDATE = %sysfunc(today(),date7.);
 %let FDATE1 = %sysfunc(today(),date9.);
 %let CUR_YEAR = %substr(&FDATE1,6,4);
  %LET CUR_CC = %substr(&FDATE1,6,2);
 %let CUR_YEAR1 = %substr(&FDATE,6,2);
  %let MM = %substr(&fdate,3,3);
  %let PREV_YEAR = %eval(&CUR_YEAR1 &operator &diff);

   %if &MM = JAN %then %let TERM_DATE = %sysfunc(catt(&dd1,&PREV_YEAR));
%ELSE
   %if &MM = FEB %then %let TERM_DATE = %sysfunc(catt(&dd2,&CUR_YEAR));
%ELSE
   %if &MM = MAR %then %let TERM_DATE = %sysfunc(catt(&dd3,&CUR_YEAR));
%ELSE
  %if &MM = APR %then %let TERM_DATE = %sysfunc(catt(&dd4,&CUR_YEAR));
%ELSE
   %if &MM = MAY %then %let TERM_DATE = %sysfunc(catt(&dd5,&CUR_YEAR));
%ELSE
  %if &MM = JUN %then %let TERM_DATE = %sysfunc(catt(&dd6,&CUR_YEAR));
%ELSE
  %if &MM = JUL %then %let TERM_DATE = %sysfunc(catt(&dd7,&CUR_YEAR));
%ELSE
  %if &MM = AUG %then %let TERM_DATE = %sysfunc(catt(&dd8,&CUR_YEAR));
%ELSE
  %if &MM = SEP %then %let TERM_DATE = %sysfunc(catt(&dd9,&CUR_YEAR));
%ELSE
  %if &MM = OCT %then %let TERM_DATE = %sysfunc(catt(&dd10,&CUR_YEAR));
%ELSE
  %if &MM = NOV %then %let TERM_DATE = %sysfunc(catt(&dd11,&CUR_YEAR));
%ELSE
  %if &MM = DEC %then %let TERM_DATE = %sysfunc(catt(&dd12,&CUR_YEAR));

 run;

%mend fdate;
%fdate;
```

```
      %put &FDATE1 &TERM_DATE &MM &PREV_YEAR &CUR_YEAR &CUR_CC;



*** ADVANCE EACH BY ONE MONTHDATE RANGE ***;
%let BEGDAT=01&MM&CUR_CC&PREV_YEAR; /* 092014 */
%let ENDDAT=&TERM_DATE; /* 092014 */

%put &BEGDAT &ENDDAT;

/*** LIMIT TO LINE OF BUSINESS ***/
%let LOB='HMO' 'COM' 'MED' 'POS' 'OPT' 'SF_HMO' 'SF_POS' 'SF_OPT' 'IFP_HMO'
'IFP_POS'
          'IFP_HMO_ACA' ;

*---------------------------*;
* Read Physican File 092014 *;
*---------------------------*;
proc import out=term_out

      datafile="&driveI.\DecisionSupport\vtaylor\SAS\ImportFiles\term.xls"
                  dbms=excel replace;
                  range='SHEET1$A1:C50';
                  getnames=YES;
                  mixed=yes;
                  scantext=yes;
                  usedate=yes;
run;
/* 092014 */
%let namelist =; /* Variable list housing names of physicans */
%let idlist =;   /* Varialbe list housing provider id of physicans */
/* 092014 */
/* Create list of phyiscan names and provider ids */
DATA term;
  set term_out;
  if code ^= 'MH' then delete;
  call symput('namelist',trim(resolve('&namelist'))||' '||trim(Last_Name));
  call symput('idlist',trim(resolve('&idlist'))||' '||trim(Provider_Id));
run;

%put &namelist &idlist;  /* 092014 */
%global namelist idlist; /* 092014 */

/* Determine number of physicans to process. Its assume each will have provider
id */
/* provide; therefore list counts should match
*/
*options symbolgen;
%macro wordcount(list);  /* 092014 */
 /* Count the number of words &list */

 %global count;
 %let count = 0;
 %do %while(%qscan(&list,&count+1,%str( )) ne %str());
```

```
      %let count = %eval(&count+1);
 %end;
 %put &count;
%mend wordcount;

%wordcount(&namelist);

%macro process(var); /* 092014 */
%global rptpath;

%let i = %eval(&var);
%put &i;

/*** ENTER PROVIDER ID OF TERMED PROVIDER ***/
%let TERMPROV = %scan(&idlist,&i); /* 092014 */ /* there may be several IDs
(CSC and QNXT)*/
%let pname = %scan(&namelist,&i);  /* 092014 */
%let pdate = &FDATE1;              /* 092014 */

%let termprovname= &pname &termprov; /* this is what goes on the printer report
*/
/*** and give the ACCESS FILES appropriate names (limited to 27 characters)
***/
%let prvname_gt13     =     &pname _>=13_&pdate; /* for mbrs without State of
WI or Medicare >=13*/
%let prvname_lt13     =      &pname _<13_&pdate; /* for mbrs without State of
WI or Medicare <13 */
%let prvname_WI_gt13  =  &pname _WI_>=13_&pdate; /* for State of Wisconsin
group only >=13       */
%let prvname_WI_lt13  =   &pname _WI_<13_&pdate; /* for State of Wisconsin
group only <13        */
%let prvname_AHS_gt13 = &pname _AHS_>=13_&pdate; /* for AHS group only >=13
*/
%let prvname_AHS_lt13 =  &pname _AHS_<13_&pdate; /* for AHS group only <13
*/
%let prvname_MED_gt13 = &pname _MED_>=13_&pdate; /* for Medicare members only
>=13 */
%let prvname_MED_lt13 =  &pname _MED_<13_&pdate; /* for Medicare members only
<13  */

data _null_;

call symput("rptpath","NHP_Provider_Terms\Continuity of
Care_Specialist\2014_Mental Health");
run;

%put &rptpath;


%CheckDirectory(&drivey\&rptpath\&termprovname);
/********* Read in Claims for timeframe & Selected Provider *************/
proc sql;
     create table PERMSAS.CLAIMS_1 as
           select     distinct
```

8

```sas
                e.claim_prov_id          as c_provid,
            e.claim_prov_first    as cdocfrst,
            e.claim_prov_last     as cdoclast,
                o.member_id

     from     star.claim_provider_dim         e,
            star.date_of_service_dim        g,
                star.product_dim               l,
            star.member_dim               o,
            star.claim_detail_fact        q

     where    "&BEGDAT"d <= g.beg_date <= "&ENDDAT"d        and
                e.claim_prov_id in ("&TERMPROV")              and
                l.lob_code       in (&LOB)                    and
                e.claim_prov_key       = q.claim_prov_key  and
            g.beg_key                   = q.beg_key         and
            l.product_key          = q.product_key      and
            o.member_id            = q.member_id         ;
quit;

/*** REMOVE ANY DUPLICATE ROWS ***/
     proc sort data = permsas.claims_1   nodupkey;
         by  member_id ;
     run;
/** force member_id to be c_mbr_ID (member_ID is already used in the
current_mbrs file ***/
     data  permsas.claims_2 (drop = member_id);
set permsas.claims_1;
c_mbr_id = member_id ;
run;

/********* Get Member Information (Unique) for Select Provider ********/
proc sql;
     create table mbr_sub_match_1 as
         select  distinct
                a.c_mbr_id,
            a.cdoclast,
            a.cdocfrst,
            a.c_provid,

                c.Subscriber_id            as s_mbr_id,
            c.sub_first_name       as s_first,
            c.sub_last_name          as s_last,
            c.sub_mid_initial        as s_MI,
            c.sub_addr1              as s_addr1,
            c.sub_addr2              as s_addr2,
            c.sub_city              as s_city,
            c.sub_state             as s_state,
            c.sub_zip               as s_zip,
                c.prod_lob_code       as  lob_code,
                c.group_association_name as group_id,

                b.family_id           as m_fam_id,
            b.member_id       as m_mbr_id,
```

9

```sas
                    b.birthdate         as m_dob,
                        b.first_name       as m_first,
                    b.last_name        as m_last,
                    b.middle_initial as m_MI,
                    b.address1          as m_addr1,
                    b.address2          as m_addr2,
                    b.city                   as m_city,
                    b.state                  as m_state,
                    b.zip_code          as m_zip,
                        (today()- b.birthdate) /365.25 as mbr_age,

                        1       as count

            from      permsas.claims_2            a,
                    PRODSAS.BI_MBRS_CURRENT_LIST        b,
                    PRODSAS.BI_SUBS_CURRENT_LIST      c

        where     a.c_mbr_id = b.member_id              and
                    b.family_id= c.sub_family_id;
quit;


/********* Eliminate Duplicates ********************************/
/*** the intent is that only one letter goes to one household ***/
proc sort data  = mbr_sub_match_1 out = mbr_sub_match_2 nodupkey;
by  m_mbr_id;
run;


/********* Print Report 1a *****************************************/
/*** THIS PROC REPORT IS FOR -STATE OF WISCONSIN- GROUPS ONLY   ***/
/***                    - MEMBERS 13 YEARS OR MORE AT TIME OF REPORT ***/
/*** once this is 'printed' to the output viewer                ***/
/*** it can then be printed onto paper and given to Group Admin ***/

options PS=50 LS=160 nodate nobyline pageno=1;

title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* STATE OF WISCONSIN GROUPS ONLY *";
title6 "* INCLUDING Members 13 years or more old (at time of this report) *";
title7 "Service Dates:  &BEGDAT - &ENDDAT";
title8 "Run Date:  &SYSDATE9.";

footnote1;

proc report data=mbr_sub_match_2
                    (WHERE=(group_id IN:('STATE OF WISCONSIN') and
                    MBR_AGE >= 13))
                    nowindows center headskip missing spacing=3 split='*';
by cdoclast;

column m_mbr_id m_last m_first m_MI m_addr1 m_city m_state m_zip count;
```

```
define m_mbr_id  /group 'Member_ID'   '--' width= 15;
define m_first   /group 'First*Name'  '--' width= 15 flow;
define m_last    /group 'Last*Name'   '--' width= 20 flow;
define m_MI      /group 'MI'          '--' width=  4;
define m_addr1   /group 'Address1'    '--' width= 30 flow;
*define m_addr2 /group 'Address2'     '--' width= 15 flow;
define m_city    /group 'City'        '--' width= 20;
define m_state   /group 'State'       '--' width=  5;
define m_zip     /group 'Zip'         '--' width=  5;
define count     /sum   'Mbr*Cnt'     '--' width=  8 format=comma8.0 ;

rbreak after / summarize dol dul;

quit;
title; footnote;

/********* Print Report 1b ****************************************/
/*** THIS PROC REPORT IS FOR -STATE OF WISCONSIN- GROUPS ONLY    ***/
/**** AND MEMBERS LESS THAN 13 YEARS OLD (at the time of the report ***/
/*** once this is 'printed' to the output viewer                ***/
/*** it can then be printed onto paper and given to Group Admin    ***/

options PS=50 LS=160 nodate nobyline pageno=1;

title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* STATE OF WISCONSIN GROUPS ONLY *";
title6 "* INCLUDING Members LESS THAN 13 years old (at time of this report) *";
title7 "Service Dates:  &BEGDAT - &ENDDAT";
title8 "Run Date:  &SYSDATE9.";

footnote1;

proc report data=mbr_sub_match_2
                   (WHERE=(group_id IN:('STATE OF WISCONSIN') and
                    MBR_AGE < 13))
                   nowindows center headskip missing spacing=3 split='*';
by cdoclast;

column   m_mbr_id  s_last  s_first  s_mi
            s_addr1   s_city  s_state  s_zip  count;

define m_mbr_id /group 'Member ID'      '--' width=15;
define s_first  /group 'Sub*First Name' '--' width=15 flow;
define s_last   /group 'Sub*Last Name'  '--' width= 20 flow;
define s_mi     /group 'Sub*MI'         '--' width=4;
define s_addr1  /group 'Sub*Address1'   '--' flow width=30;
define s_city   /group 'Sub* City'      '--' width=20;
define s_state  /group 'Sub**State'     '--' width=5;
define s_zip    /group 'Sub*Zip'        '--' width=5;
define count    /sum   'Sub*Count'      '--' format=comma8.0 width=8;
```

```sas
rbreak after / summarize dol dul;
quit;
title; footnote;

/********* Print Report 1a *****************************************/
/*** THIS PROC REPORT IS FOR -AFFINITY HEALTH SYSTEM- GROUPS ONLY   ***/
/***                   - MEMBERS 13 YEARS OR MORE AT TIME OF REPORT  ***/
/*** once this is 'printed' to the output viewer                   ***/
/*** it can then be printed onto paper and given to Group Admin    ***/

options PS=50 LS=160 nodate nobyline pageno=1;

title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* AFFINITY HEALTH SYSTEM GROUPS ONLY *";
title6 "* INCLUDING Members 13 years or more old (at time of this report) *";
title7 "Service Dates:  &BEGDAT - &ENDDAT";
title8 "Run Date:  &SYSDATE9.";

footnote1;

proc report data=mbr_sub_match_2
                (WHERE=(group_id IN:('AFFINITY HEALTH SYSTEM', 'AFFINITY
HEALTH SYSTEM INC', 'AFFINITY HEALTH SYSTEM INC SF', 'MINISTRY HEALTH CARE EAST
REGION SF') and
                MBR_AGE >= 13))
                nowindows center headskip missing spacing=3 split='*';
by cdoclast;

column m_mbr_id m_last m_first m_MI m_addr1 m_city m_state m_zip count;

define m_mbr_id  /group 'Member_ID'  '--' width= 15;
define m_first   /group 'First*Name' '--' width= 15 flow;
define m_last    /group 'Last*Name'  '--' width= 20 flow;
define m_MI      /group 'MI'         '--' width=  4;
define m_addr1   /group 'Address1'   '--' width= 30 flow;
*define m_addr2 /group 'Address2'    '--' width= 15 flow;
define m_city   /group 'City'        '--' width= 20;
define m_state  /group 'State'       '--' width=  5;
define m_zip    /group 'Zip'         '--' width=  5;
define count    /sum   'Mbr*Cnt'     '--' width=  8 format=comma8.0 ;

rbreak after / summarize dol dul;

quit;
title; footnote;

/********* Print Report 1b *****************************************/
/*** THIS PROC REPORT IS FOR -AFFINITY HEALTH SYSTEM- GROUPS ONLY   ***/
/**** AND MEMBERS LESS THAN 13 YEARS OLD (at the time of the report ***/
/*** once this is 'printed' to the output viewer                   ***/
/*** it can then be printed onto paper and given to Group Admin    ***/
```

```
options PS=50 LS=160 nodate nobyline pageno=1;

title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* AFFINITY HEALTH SYSTEM GROUPS ONLY *";
title6 "* INCLUDING Members LESS THAN 13 years old (at time of this report) *";
title7 "Service Dates:  &BEGDAT - &ENDDAT";
title8 "Run Date:  &SYSDATE9.";

footnote1;

proc report data=mbr_sub_match_2
                    (WHERE=(group_id IN:('AFFINITY HEALTH SYSTEM', 'AFFINITY
HEALTH SYSTEM INC', 'AFFINITY HEALTH SYSTEM INC SF', 'MINISTRY HEALTH CARE EAST
REGION SF') and
                    MBR_AGE < 13))
                    nowindows center headskip missing spacing=3 split='*';
by cdoclast;

column   m_mbr_id  s_last  s_first  s_mi
           s_addr1   s_city  s_state  s_zip  count;

define m_mbr_id /group 'Member ID'       '--' width=15;
define s_first  /group 'Sub*First Name' '--' width=15 flow;
define s_last   /group 'Sub*Last Name'  '--' width= 20 flow;
define s_mi      /group 'Sub*MI'         '--' width=4;
define s_addr1  /group 'Sub*Address1'   '--' flow width=30;
define s_city   /group 'Sub* City'      '--' width=20;
define s_state  /group 'Sub**State'     '--' width=5;
define s_zip    /group 'Sub*Zip'        '--' width=5;
define count    /sum   'Sub*Count'      '--' format=comma8.0 width=8;

rbreak after / summarize dol dul;
quit;
title; footnote;
/******************** Print Report 2A ***************************/
/*** THIS PROC REPORT IS FOR                              ***/
/*** NON-STATE OF WISCONSIN GROUPS-                       ***/
/*** AND NON MEDICARE                                     ***/
/***  AND MEMBERS 13 YEARS OR MORE AT TIME OF REPORT      ***/
/*** once this is 'printed' to the output viewer          ***/
/*** it can then be printed onto paper and given to Group Admin  ***/

options PS=50 LS=160 nodate nobyline pageno=1;

title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* EXCLUDING STATE OF WISCONSIN AND AFFINITY HEALTH SYSTEM GROUPS *";
title6 "* EXCLUDING MEDICARE LINE OF BUSINESS *";
```

```
title7 "* INCLUDING Members 13 years or more old (at time of this report) *";
title8 "Service Dates:  &BEGDAT - &ENDDAT";
title9 "Run Date:  &SYSDATE9.";

footnote1;

proc report data=mbr_sub_match_2 (WHERE=(group_id NOT IN:('STATE OF WISCONSIN',
'AFFINITY HEALTH SYSTEM', 'AFFINITY HEALTH SYSTEM INC', 'AFFINITY HEALTH SYSTEM
INC SF', 'MINISTRY HEALTH CARE EAST REGION SF')
                                        AND LOB_CODE NOT IN ('MED')
                                        AND MBR_AGE >= 13))

     nowindows center headskip missing spacing=3 split='*';
by cdoclast;

column m_mbr_id m_last m_first m_MI m_addr1 m_city m_state m_zip count;

define m_mbr_id  /group 'Member_ID'  '--' width= 15;
define m_first   /group 'First*Name' '--' width= 15 flow;
define m_last    /group 'Last*Name'  '--' width= 20 flow;
define m_MI       /group 'MI'         '--' width=  4;
define m_addr1   /group 'Address1'   '--' width= 30 flow;
*define m_addr2 /group 'Address2'    '--' width= 15 flow;
define m_city   /group 'City'        '--' width= 20;
define m_state  /group 'State'       '--' width=  5;
define m_zip     /group 'Zip'         '--' width=  5;
define count    /sum   'Mbr*Cnt'     '--' width=  8 format=comma8.0 ;
rbreak after / summarize dol dul;

quit;

/******************** Print Report 2B ********************************/
/*** THIS PROC REPORT IS FOR                              ***/
/*** NON-STATE OF WISCONSIN GROUPS-                       ***/
/*** AND NON MEDICARE                                     ***/
/**** AND MEMBERS LESS THAN 13 YEARS OLD (at the time of the report  ***/
/*** once this is 'printed' to the output viewer          ***/
/*** it can then be printed onto paper and given to Group Admin    ***/

options PS=50 LS=160 nodate nobyline pageno=1;

title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* EXCLUDING STATE OF WISCONSIN AND AFFINITY HEALTH SYSTEM GROUPS *";
title6 "* EXCLUDING MEDICARE LINE OF BUSINESS *";
title7 "* INCLUDING Members LESS THAN 13 years old (at time of this report) *";
title8 "Service Dates:  &BEGDAT - &ENDDAT";
title9 "Run Date:  &SYSDATE9.";

footnote1;
```

```sas
proc report data=mbr_sub_match_2 (WHERE=(group_id NOT IN:('STATE OF WISCONSIN',
'AFFINITY HEALTH SYSTEM', 'AFFINITY HEALTH SYSTEM INC', 'AFFINITY HEALTH SYSTEM
INC SF', 'MINISTRY HEALTH CARE EAST REGION SF')
                                         AND LOB_CODE NOT IN ('MED')
                                         AND MBR_AGE < 13))

      nowindows center headskip missing spacing=3 split='*';
by cdoclast;

column   m_mbr_id  s_last   s_first  s_mi
            s_addr1    s_city  s_state  s_zip  count;

define m_mbr_id /group 'Member ID'      '--' width=15;
define s_first  /group 'Sub*First Name' '--' width=15 flow;
define s_last   /group 'Sub*Last Name'  '--' width= 20 flow;
define s_mi     /group 'Sub*MI'         '--' width=4;
define s_addr1  /group 'Sub*Address1'   '--' flow width=30;
define s_city   /group 'Sub* City'      '--' width=20;
define s_state  /group 'Sub**State'     '--' width=5;
define s_zip    /group 'Sub*Zip'        '--' width=5;
define count    /sum   'Sub*Count'      '--' format=comma8.0 width=8;

rbreak after / summarize dol dul;
quit;
title; footnote;


/********* Print Report 3A ***************************************/
/*** THIS PROC REPORT IS FOR- MEDICARE LINE OF BUSINESS -ONLY   ***/
/***  AND MEMBERS 13 YEARS OR MORE AT TIME OF REPORT            ***/
/*** once this is 'printed' to the output viewer               ***/
/*** it can then be printed onto paper and given to Group Admin ***/

options PS=50 LS=160 nodate nobyline pageno=1;

title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* MEDICARE LINE OF BUSINESS ONLY *";
title6 "* INCLUDING Members 13 years or more old (at time of this report) *";
title7 "Service Dates:  &BEGDAT - &ENDDAT";
title8 "Run Date:  &SYSDATE9.";

footnote1;

proc report data=mbr_sub_match_2 (WHERE=(group_id NOT IN:('STATE OF WISCONSIN')
                                    AND LOB_CODE IN ('MED')
                                    AND MBR_AGE >= 13))
      nowindows center headskip missing spacing=3 split='*';
by cdoclast;

column m_mbr_id m_last m_first m_MI m_addr1 m_city m_state m_zip count;
```

```
define m_mbr_id  /group 'Member_ID'   '--' width= 15;
define m_first   /group 'First*Name' '--' width= 15 flow;
define m_last    /group 'Last*Name'   '--' width= 20 flow;
define m_MI       /group 'MI'          '--' width=  4;
define m_addr1   /group 'Address1'    '--' width= 30 flow;
*define m_addr2 /group 'Address2'     '--' width= 15 flow;
define m_city   /group 'City'          '--' width= 20;
define m_state  /group 'State'         '--' width=  5;
define m_zip     /group 'Zip'           '--' width=  5;
define count    /sum    'Mbr*Cnt'      '--' width=  8 format=comma8.0 ;


rbreak after / summarize dol dul;


quit;


/********* Print Report 3B *****************************************/
/*** THIS PROC REPORT IS FOR- MEDICARE LINE OF BUSINESS -ONLY      ***/
/**** AND MEMBERS LESS THAN 13 YEARS OLD (at the time of the report ***/
/*** once this is 'printed' to the output viewer                   ***/
/*** it can then be printed onto paper and given to Group Admin    ***/


options PS=50 LS=160 nodate nobyline pageno=1;


title1 "Active NHP Members who have Recieved Services";
title2 "from Specialists who are Terming";
title3 "*** &termprovname ***";
title4 "*** Subscriber Level ***";
title5 "* MEDICARE LINE OF BUSINESS ONLY *";
title6 "* INCLUDING Members LESS THAN 13 years old (at time of this report) *";
title7 "Service Dates:  &BEGDAT - &ENDDAT";
title8 "Run Date:  &SYSDATE9.";


footnote1;


proc report data=mbr_sub_match_2 (WHERE=(group_id NOT IN:('STATE OF WISCONSIN')
                                  AND LOB_CODE IN ('MED')
                                  AND MBR_AGE < 13))
     nowindows center headskip missing spacing=3 split='*';
by cdoclast;


column   m_mbr_id  s_last  s_first   s_mi
          s_addr1    s_city  s_state  s_zip   count;


define m_mbr_id /group 'Member ID'      '--' width=15;
define s_first  /group 'Sub*First Name' '--' width=15 flow;
define s_last   /group 'Sub*Last Name'  '--' width= 20 flow;
define s_mi      /group 'Sub*MI'         '--' width=4;
define s_addr1  /group 'Sub*Address1'   '--' flow width=30;
define s_city   /group 'Sub* City'       '--' width=20;
define s_state  /group 'Sub**State'      '--' width=5;
define s_zip     /group 'Sub*Zip'        '--' width=5;
define count    /sum    'Sub*Count'      '--' format=comma8.0 width=8;


rbreak after / summarize dol dul;
```

```
quit;

/********* Format Data ***************************************/
data docout_1 ;
set mbr_sub_match_2;

label       cdoclast = Provider Last
        cdocfrst = Provider First
        c_provid = Provider ID
           c_mbr_id = Claim_Member_ID

           lob_code = Line of Business
           group_id = Member Group ID
           m_fam_id = Family ID
        mbr_age   = Member Age
        m_mbr_id = Member ID
           m_first  = Member First
        m_last   = Member Last
        m_MI      = Member MI
        m_addr1   = Member Address1
        m_addr2   = Member Address2
        m_city   = Member City
        m_state  = Member State
        m_zip     = Member Zip

           s_mbr_id = Subscriber ID
           s_first  = Sub First Name
        s_last   = Sub Last Name
        s_MI      = Sub MI
        s_addr1  = Sub Address1
        s_addr2  = Sub Address2
        s_city   = Sub City
        s_state  = Sub State
        s_zip    = Sub Zip

           count     = Count ;
      run;

/********* Eliminate Duplicates ***************************/
proc sort data = docout_1
             out = docout_2  nodupkey;
by   m_mbr_id   ;
run;



data term_prov_STATE_of_WIGE13;
set  docout_2;
if   group_id = 'STATE OF WISCONSIN'  and
          mbr_age >= 13 ;
run;
data term_prov_ahsGE13;
set  docout_2;
```

```
if   group_id in ('AFFINITY HEALTH SYSTEM', 'AFFINITY HEALTH SYSTEM INC'
'AFFINITY HEALTH SYSTEM INC SF' 'MINISTRY HEALTH CARE EAST REGION SF')and
         mbr_age >= 13;
run;
data term_prov_COMMERCIALGE13;
set  docout_2;
where   group_id ^IN ('STATE OF WISCONSIN', 'AFFINITY HEALTH SYSTEM', 'AFFINITY
HEALTH SYSTEM INC' 'AFFINITY HEALTH SYSTEM INC SF' 'MINISTRY HEALTH CARE EAST
REGION SF')and
         mbr_age >= 13;
run;
data term_prov_STATE_of_WILT13;
set  docout_2;
if   group_id = 'STATE OF WISCONSIN'  and
         mbr_age < 13 ;
run;
data term_prov_ahsLT13;
set  docout_2;
if   group_id in ('AFFINITY HEALTH SYSTEM', 'AFFINITY HEALTH SYSTEM INC'
'AFFINITY HEALTH SYSTEM INC SF' 'MINISTRY HEALTH CARE EAST REGION SF')and
         mbr_age < 13;
run;
data term_prov_COMMERCIALLT13;
set  mbr_sub_match_2;
where   group_id ^IN ('STATE OF WISCONSIN', 'AFFINITY HEALTH SYSTEM', 'AFFINITY
HEALTH SYSTEM INC' 'AFFINITY HEALTH SYSTEM INC SF' 'MINISTRY HEALTH CARE EAST
REGION SF')and
         mbr_age < 13;
run;


libname comge13 excel "&drivey\&rptpath\&termprovname\&termprovname COMMERCIAL
ge 13.xls" ver=2002 ;
libname SOWge13 excel "&drivey\&rptpath\&termprovname\&termprovname State of WI
ge 13.xls" ;
libname AHSge13 excel "&drivey\&rptpath\&termprovname\&termprovname MINISTRY ge
13.xls" ;
libname comlt13 excel "&drivey\&rptpath\&termprovname\&termprovname COMMERCIAL
lt 13.xls" ver=2002 ;
libname SOWlt13 excel "&drivey\&rptpath\&termprovname\&termprovname State of WI
lt 13.xls" ;
libname AHSlt13 excel "&drivey\&rptpath\&termprovname\&termprovname MINISTRY lt
13.xls" ;

data comge13.CommercialGE13;
set work.TERM_PROV_COMMERCIALGE13;
run;
data sowge13.WIGE13;
set work.TERM_PROV_State_of_WIGE13;
run;
data ahsge13.MINGE13;
set work.TERM_PROV_ahsGE13;
run;
data comlt13.CommercialLT13;
```

```
set work.TERM_PROV_COMMERCIALLT13;
run;
data sowlt13.WILT13;
set work.TERM_PROV_State_of_WILT13;
run;
data ahslt13.MINLT13;
set work.TERM_PROV_ahsLT13;
run;

libname comge13 clear;
libname SOWge13 clear;
libname AHSge13 clear;
libname comlt13 clear;
libname SOWlt13 clear;
libname AHSlt13 clear;

/* clean up */
proc datasets library=work nolist;
   delete TERM_PROV_COMMERCIALGE13 TERM_PROV_State_of_WIGE13 TERM_PROV_ahsGE13
TERM_PROV_COMMERCIALLT13
         TERM_PROV_State_of_WILT13 TERM_PROV_ahsLT13;
run;

%mend process; /* Terminates program reiteration */

data _null_; /* 092014 */

  cnt = symget('count');

  do j = 1 to cnt;
     call execute ('%process('||j||')');
  end;

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 time12.5;
run;
```

Terminating Providers

SCRIPT FOR TERM PROVIDER PANEL

1. Kimberly Korb will populate the **term.xls** spreadsheet with provider's LAST_NAME, PROVIDER_Id and Code. Code being two characters, MH=Mental Health PC=Primary Care, SP=Specialty

2. Kimberly will then email the spreadsheet to vtaylor@net.com and jresch@net.com

3. Vincent or Jennifer will save spreadsheet at I:\DecisionSupport\vtaylor\sas\importfiles

4. Go to I:\DecsionSupport\RoutineReports\OnDemand\Terminating_Provider_Lists\Program\SAS

5. Open and execute program(s) Mental_Health_Providers Rev7 SEP17, Non-Mental Health rev9 SEP16, or PCP terms Ver7 SEP16

6. Go to Y:\NHP_Provider_Terms\Continuity of Care_Specialist check 2014_Specialist, 2014_Mental_Health folder. Print out spreadsheet using landscape style

7. Go to Y:\NHP_Provider_Terms\Continuity of Care_PCP check 2014_PCP folder. Print out spreadsheet using landscape style

8. Send Kimberly Korb a mail informing her of providers with zero data.

Mentalhealth_term.xls (used as an example)

Last_Name	Provider_ID
Peterson	P00VVVVV
Garstka	P00XXXXX
Riedl	P00ZZZZZZ

NOTE: If there are errors in the log; check the RPTPATH variable to see if it translated correctly, if not re-run the program from top. The macro variable RPTPATH is passed to the macro 'CheckDirectory' which is invoked from within the macro 'Process". There are times in such situations that macro variables will not be written to the macro variable pool on the first pass.