/\*

Project Title: ACA Chart Project Target List CY2023

Requesting Department: Internal

Requestor: XXXXXXXXXX

Origination Date: 12/5/2023

Requested Completion Date: 12/11/2023

Date received: 12/5/2023

Assigned Priority: High

Assigned Delivery Date: 12/11/2023

RDA\_Project\_number: XXXXXXXXXX

Assigned Analyst: Meagan Windler

Assigned Days: 2

Assigned Completion Date: 12/11/2023

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Setup \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%Let rundate= %sysfunc(today(), YYMMDDN8.); \*Date for file name;

%Let RDA=XXXXXXXXXX;

%Let Root=\\XXXXXXXXXX;

%Let Analyst=MW;

%Let Outpath=&Root\XXXXXXXXXX\ ACA Chart Project Target List CY2023\Output;

%Let reportdate = %sysfunc(today(), date9.);

/\* Current time in HHMMSS format, removing colons \*/

%let timepart = %sysfunc(putn(%sysfunc(time()), time8.));

%let cleantimepart = %sysfunc(translate(&timepart, , :));

/\* Concatenate the date and cleaned time parts for the timestamp \*/

%let timestamp = &rundate.%sysfunc(compress(&timepart,":"));

%put &timestamp;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Import files from Change Healthcare \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*Import Chase List raw text file from Chase List Creator portal on 12/6/2023;

PROC IMPORT DATAFILE="&Root\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Data\1260-CHART\_UI\_OPTIMIZED-2023120615274085.txt"

OUT=work.chaselistraw

DBMS=DLM

REPLACE;

DELIMITER='|'; /\* Set the delimiter as pipe \*/

GETNAMES=YES; /\* The first row contains variable names \*/

GUESSINGROWS=32767; /\* Increase the number of rows SAS scans to determine variable lengths \*/

RUN;

\*Import All Members file with score and PCP infofrom Risk View Commercial;

PROC IMPORT OUT= work.memberlist

DATAFILE= "&Root\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Data\members\_20231206100007.xlsx"

DBMS=EXCEL

REPLACE;

SHEET="Sheet0";

GETNAMES=YES; /\* The first row contains variable names \*/

MIXED=NO;

RUN;

\*Add member list score and PCP information to chase list;

PROC SQL;

CREATE TABLE chaselist2 AS

SELECT DISTINCT a.\*,

b.current\_score,

b.probable\_score,

b.max\_potential\_score,

b.Diff\_Probable\_\_\_Current,

b.Diff\_Potential\_\_\_Current,

b.pcp\_provider\_Number,

b.pcp\_first\_name,

b.pcp\_last\_name

FROM chaselistraw AS a

INNER JOIN memberlist AS b ON a.member\_Number = b.member\_Number;

QUIT;

\*Add strtdate column & Change PCP\_Provider\_Number to Aff\_ to add Amisys IRS info \*;

Data chaselist3;

Set chaselist2;

strtdate = today();

format strtdate MMDDYY10.;

AFF\_ = PCP\_Provider\_Number; Run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Add IRS info from Amisys \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*Use attachaffinfo macro to add IRS\_ info to chaselist3 table;

%attachaffinfo(chaselist3,irs\_);

\*Add IRSname info using IRSname format;

Data chaselist3;

Set CHASELIST3\_WITHINFO;

IRS\_Name= put(IRS\_,$irsname.); Run;

\*Drop empty fields and reorder;

Data chaselist\_aca\_2023;

Set CHASELIST3\_WITHINFO (Drop=LOB\_Member\_ID Member\_CMS\_County\_Code Member\_County Member\_Population\_Group Provider\_Email cnt hit aff); Run;

\*Save a copy of the final working chase list to outpath folder;

Libname outpath "&Outpath";

Data outpath.chaselist\_aca\_2023\_&rundate;

Set chaselist\_aca\_2023; Run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 01/25/2024 Updates to final list

Include CCOK Flag in optional fields\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*Import Chase List .txt file that was converted into .xlsx file from Chase List Creator portal on 01/25/24;

/\* Import the Excel file \*/

PROC IMPORT DATAFILE="&Root\HDA\RDA\mwindler\RDA20231205006 ACA Chart Project Target List CY2023\Data\1260-CHART\_OPTIMIZED\_STANDARD-2024012519520012v2.xlsx"

OUT=work.chaselistraw3 /\* Specify the output SAS dataset name \*/

DBMS=XLSX REPLACE; /\* Specify the DBMS as XLSX for Excel files \*/

/\* Specify the sheet name or number from the Excel file \*/

SHEET="1260-CHART\_OPTIMIZED\_STANDARD-2"; /\* Replace with your sheet name or number \*/

/\* Specify options for the Excel file \*/

GETNAMES=YES; /\* The first row contains variable names \*/

/\* Specify the format and length for Provider\_Number as character \*/

INFORMAT Provider\_Number $CHAR16.;

RUN;

\*read in final list from Angela;

libname final '\\tXXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Output';

data final0;

set final.finallist (Drop=Provider\_Number); run;

\*add in chase weight, CMS specialty code, and POS code from raw chaselist;

proc sql;

create table final1 as

select a.\*,

b.chase\_weight,

b.Provider\_cms\_specialty\_code,

b.Provider\_First\_Name,

b.Provider\_Last\_Name,

b.Provider\_Number,

b.pos as POS\_CODE

from final0 as a

inner join chaselistraw3 as b

on a.chart\_id = b.chart\_id;

quit;

data final2;

Set final1 (Drop=POS); run;

\*adding in project name & renaming columns;

data final3;

set final2 (rename=(

member\_number=MEMBER

Provider\_Number=PROVIDER

chase\_weight=CHASE\_WEIGHT

provider\_group\_number=PROVIDER\_GROUP

Provider\_First\_Name=PROVIDER\_FIRST\_NAME

Provider\_Last\_Name=PROVIDER\_LAST\_NAME

Provider\_cms\_specialty\_code=CMS\_SPECIALTY\_CODE

chart\_rank=CHASE\_LIST\_RANK

provider\_address\_1=CHART\_ADDRESS1

provider\_address\_2=CHART\_ADDRESS2

provider\_city=CHART\_CITY

provider\_state=CHART\_STATE

provider\_zip=CHART\_ZIP

provider\_phone=PROVIDER\_PHONE

provider\_fax=PROVIDER\_FAX

ccokflag=CCOK\_FLAG

POS\_CODE=POS

));

PROJECT\_NAME= 'XXXXXXXXXX';

CHART\_IMAGE\_NAME='';

PROVIDER\_EMAIL='';

SPECIAL\_HANDLING\_CATEGORY='';

SPECIAL\_HANDLING\_COMMENT='';

CONTACT\_NAME='';

CONTACT\_PHONE='';

CONTACT\_EXT='';

CONTACT\_FAX='';

USER\_FIELD\_3='';

If CCOK\_FLAG = 0 then CHART\_CHASE\_METHOD = 'XXXXXXXXXX';

else CHART\_CHASE\_METHOD = 'XXXXXXXXXX';

If CCOK\_FLAG = 0 then IS\_ABSTRACT = 'N';

else IS\_ABSTRACT = 'Y';

run;

data final4\_&timestamp;

retain PROJECT\_NAME MEMBER PROVIDER IS\_ABSTRACT CHART\_IMAGE\_NAME

CHASE\_WEIGHT PROVIDER\_GROUP CHASE\_LIST\_RANK CCOK\_FLAG CHART\_CHASE\_METHOD

USER\_FIELD\_3 PROVIDER\_FIRST\_NAME PROVIDER\_LAST\_NAME CMS\_SPECIALTY\_CODE

CHART\_ADDRESS1 CHART\_ADDRESS2 CHART\_CITY CHART\_STATE CHART\_ZIP

PROVIDER\_PHONE PROVIDER\_FAX PROVIDER\_EMAIL POS SPECIAL\_HANDLING\_CATEGORY

SPECIAL\_HANDLING\_COMMENT CONTACT\_NAME CONTACT\_PHONE CONTACT\_EXT CONTACT\_FAX;

set final3 (Keep= PROJECT\_NAME MEMBER PROVIDER IS\_ABSTRACT CHART\_IMAGE\_NAME

CHASE\_WEIGHT PROVIDER\_GROUP CHASE\_LIST\_RANK CCOK\_FLAG CHART\_CHASE\_METHOD

USER\_FIELD\_3 PROVIDER\_FIRST\_NAME PROVIDER\_LAST\_NAME CMS\_SPECIALTY\_CODE

CHART\_ADDRESS1 CHART\_ADDRESS2 CHART\_CITY CHART\_STATE CHART\_ZIP

PROVIDER\_PHONE PROVIDER\_FAX PROVIDER\_EMAIL POS SPECIAL\_HANDLING\_CATEGORY

SPECIAL\_HANDLING\_COMMENT CONTACT\_NAME CONTACT\_PHONE CONTACT\_EXT CONTACT\_FAX);

if member = 'XXXXXXXXXX' and chase\_list\_rank= 1904 then delete;

if member = 'XXXXXXXXXX' and chase\_list\_rank= 3449 then delete; run;

\*Save final SAS datasest to outpath;

%let Root=\\XXXXXXXXXX;

%let Outpath=&Root\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Output;

libname outlib "&Outpath";

/\* Copy the dataset to the specified directory \*/

data outlib.FINAL4\_20240131153531;

set WORK.FINAL4\_20240131153531;

run;

\*export final file as a .csv;

%let Root=\\XXXXXXXXXX;

%let Outpath=&Root\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Output;

%let Filename=FINAL4\_20240131153531.csv; /\* Define the output filename \*/

/\* Use a DATA step to write the dataset as a CSV file with | as the delimiter \*/

filename csvout "&Outpath.\&Filename.";

data \_null\_;

set WORK.FINAL4\_20240131153531;

file csvout dlm='|' dsd lrecl=32767;

put (\_all\_) (:);

run;

filename csvout clear;

\*exporting as an Excel file;

proc export data=WORK.FINAL4\_20240131153531

outfile="&Outpath.\FINAL4\_20240131153531.xlsx"

dbms=xlsx

replace;

run;

\*Investigating duplicates;

data final0;

set final.finallist; run;

/\*proc freq data=final0;\*/

/\* tables chart\_ID\*member\_Number\*Provider\_NPI / noprint out=duplicate\_records(keep=chart\_ID member\_Number Provider\_NPI count);\*/

/\*run;\*/

/\*\*/

/\*proc freq data=final0;\*/

/\* tables member\_Number\*Provider\_NPI / noprint out=duplicate\_records2 (keep=chart\_ID member\_Number Provider\_NPI count);\*/

/\*run;\*/

/\*\*/

/\*proc print data=duplicate\_records;\*/

/\*run;\*/

/\*\*/

/\*proc print data= duplicate\_records2;\*/

/\*run;\*/

proc sort data=FINAL4\_20240131153531;

by member provider; run;

data duplicate\_records3;

set FINAL4\_20240131153531;

by member provider;

if first.provider then count = 0;

count + 1;

if last.provider and count > 1 then output;

run;

proc sort data=duplicate\_records3;

by descending count;

proc print data=duplicate\_records3;

run;

\*identify duplicate record lines;

data duplicates\_final;

Set FINAL4\_20240131111012;

where member = 'XXXXXXXXXX' or member = 'XXXXXXXXXX'; run;

\*exporting duplicates as an Excel file;

proc export data=WORK.duplicate\_records3

outfile="&Outpath.\duplicate\_records3.xlsx"

dbms=xlsx

replace;

run;

proc export data=WORK.duplicate\_records2

outfile="&Outpath.\duplicate\_records2.xlsx"

dbms=xlsx

replace;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*Checking final totals for Angela;\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let Root=\\XXXXXXXXXX;

%let Outpath=&Root\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Output;

libname outlib "&Outpath";

/\* Copy the dataset to the specified directory \*/

data work.final4\_20240131153531;

set outlib.final4\_20240131153531;

run;

/\* Creating a dataset with unique members- Count should be 3860; Check shows 3862\*/

proc sort data=work.final4\_20240131153531 out=unique\_members nodupkey;

by MEMBER;

run;

/\* Count unique members - Count should be 3860\*/

proc sql;

select count(\*) as UniqueMemberCount from unique\_members;

quit;

/\* Creating a dataset with unique providers - Count should be 2636; check shows 2637\*/

proc sort data=work.final4\_20240131153531 out=unique\_providers nodupkey;

by PROVIDER;

run;

/\* Count unique providers - Count should be 2636\*/

proc sql;

select count(\*) as UniqueProviderCount from unique\_providers;

quit;

/\* Creating a dataset with unique addresses - Count should be 378; check shows 868\*/

proc sort data=work.final4\_20240131153531 out=unique\_addresses nodupkey;

by CHART\_ADDRESS1 CHART\_ADDRESS2 CHART\_CITY CHART\_STATE CHART\_ZIP;

run;

/\* Count unique addresses - Count should be 378 \*/

proc sql;

select count(\*) as UniqueAddressCount from unique\_addresses;

quit;

\*address count is off, trying something different;

/\* Remove leading/trailing whitespace and standardize the case \*/

data work.cleaned\_addresses;

set work.final4\_20240131153531;

length Cleaned\_Address $200.; /\* Adjust the length accordingly \*/

Cleaned\_Address = catx(' ',

lowcase(strip(CHART\_ADDRESS1)),

lowcase(strip(CHART\_ADDRESS2)),

lowcase(strip(CHART\_CITY)),

lowcase(strip(CHART\_STATE)),

strip(CHART\_ZIP)

);

/\* Handle missing values as blanks to ensure they don't create artificial uniqueness \*/

if Cleaned\_Address = ' ' then Cleaned\_Address = 'Missing';

run;

/\* Deduplicate based on the cleaned and combined address \*/

proc sort data=work.cleaned\_addresses out=unique\_addresses nodupkey;

by Cleaned\_Address;

run;

/\* Verify the count, count should be 378; check shows 867 \*/

proc sql;

select count(\*) as UniqueAddressCount

from unique\_addresses;

quit;

proc sql;

/\* Unique Member/Provider Count (combining MEMBER and PROVIDER)- Count should be 5635, check shows 5637\*/

select count(distinct catx(' ', MEMBER, PROVIDER)) as UniqueMemberProviderCount

from work.final4\_20240131153531;

/\* Unique Address/Provider Phone Count (combining CHART\_ADDRESS1 and PROVIDER\_PHONE)- Count should be 407, check shows 882\*/

select count(distinct catx(' ', CHART\_ADDRESS1, PROVIDER\_PHONE)) as UniqueAddressProviderPhoneCount

from work.final4\_20240131153531;

/\* Total Active Chases in Sample (total rows in the dataset)- Count should be 5635, check shows 5637 \*/

select count(\*) as TotalActiveChasesInSample

from work.final4\_20240131153531;

/\* Abs Only Chase Count (rows where IS\_ABSTRACT = 'Y')- Count should be 2634, check shows 2635 \*/

select count(\*) as AbsOnlyChaseCountY

from work.final4\_20240131153531

where IS\_ABSTRACT = 'Y';

/\* Missing Provider First Name (count of missing or blank PROVIDER\_FIRST\_NAME) - Count should be 1046, check shows 1046\*/

select count(\*) as MissingProviderFirstName

from work.final4\_20240131153531

where PROVIDER\_FIRST\_NAME is missing or PROVIDER\_FIRST\_NAME = ' ';

/\* Missing Provider Group (count of missing or blank PROVIDER\_GROUP) - Count should be 956, check shows 956 \*/

select count(\*) as MissingProviderGroup

from work.final4\_20240131153531

where PROVIDER\_GROUP is missing or PROVIDER\_GROUP = ' ';

/\* Missing Provider Fax (count of missing or blank PROVIDER\_Fax) - Count should be 8, check shows 10 \*/

select count(\*) as MissingProviderFax

from work.final4\_20240131153531

where PROVIDER\_FAX is missing;

quit;

\*adding NPI to final chaselist;

PROC IMPORT DATAFILE="&Root\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Data\1260-CHART\_OPTIMIZED\_STANDARD-2024012519520012v2.xlsx"

OUT=work.chaselistraw3 /\* Specify the output SAS dataset name \*/

DBMS=XLSX REPLACE; /\* Specify the DBMS as XLSX for Excel files \*/

/\* Specify the sheet name or number from the Excel file \*/

SHEET="1260-CHART\_OPTIMIZED\_STANDARD-2"; /\* Replace with your sheet name or number \*/

/\* Specify options for the Excel file \*/

GETNAMES=YES; /\* The first row contains variable names \*/

/\* Specify the format and length for Provider\_Number as character \*/

INFORMAT Provider\_Number $CHAR16.;

RUN;

\*read in final list from Angela;

libname final '\\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Output';

data final0;

set final.finallist (Drop=Provider\_Number); run;

/\* Missing Provider\_NPI (count of missing or blank PROVIDER\_NPI) - Count should be 864,check shows 951 \*/

proc sql;

create table work.final6 as

select a.\*,

b.Provider\_NPI

from work.final0 as a

inner join work.chaselistraw3 as b

on a.chart\_id = b.chart\_id

where b.Provider\_NPI is missing;

quit;

proc sql;

/\* Missing Provider\_NPI (count of missing or blank PROVIDER\_NPI) - Count should be 864, check shows 951\*/

select count(\*) as MissingProviderNPI

from work.final6

where Provider\_NPI is missing; quit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 02/12/2024 Splitting the list & adding in crosswalked fields\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%Let rundate= %sysfunc(today(), YYMMDDN8.); \*Date for file name;

%Let RDA=XXXXXXXXXX;

%Let Root=\\XXXXXXXXXX;

%Let Analyst=MW;

%Let Outpath=&Root\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Output;

%Let reportdate = %sysfunc(today(), date9.);

/\* Current time in HHMMSS format, removing colons \*/

%let timepart = %sysfunc(putn(%sysfunc(time()), time8.));

%let cleantimepart = %sysfunc(translate(&timepart, , :));

/\* Concatenate the date and cleaned time parts for the timestamp \*/

%let timestamp = &rundate.%sysfunc(compress(&timepart,":"));

%put &timestamp;

\*Importing original raw chaselist, our final list, and our final chaselist we submitted to Change on 1/31/24;

PROC IMPORT DATAFILE="&Root\HDA\RDA\mwindler\RDA20231205006 ACA Chart Project Target List CY2023\Data\1260-CHART\_OPTIMIZED\_STANDARD-2024012519520012v2.xlsx"

OUT=work.chaselistraw3 /\* Specify the output SAS dataset name \*/

DBMS=XLSX REPLACE; /\* Specify the DBMS as XLSX for Excel files \*/

/\* Specify the sheet name or number from the Excel file \*/

SHEET="1260-CHART\_OPTIMIZED\_STANDARD-2"; /\* Replace with your sheet name or number \*/

/\* Specify options for the Excel file \*/

GETNAMES=YES; /\* The first row contains variable names \*/

/\* Specify the format and length for Provider\_Number as character \*/

INFORMAT Provider\_Number $CHAR16.;

RUN;

\*read in final list from Angela that includes all data fields;

libname final '\\XXXXXXXXXX\XXXXXXXXXX ACA Chart Project Target List CY2023\Output';

data orig\_final\_list;

set final.finallist (Drop=Provider\_Number); run;

\*creating a duplicate final\_list that includes chart\_ID for referencing;

proc sql;

create table final1 as

select a.\*,

b.chase\_weight,

b.Provider\_cms\_specialty\_code,

b.Provider\_First\_Name,

b.Provider\_Last\_Name,

b.Provider\_Number,

b.pos as POS\_CODE

from orig\_final\_list as a

inner join chaselistraw3 as b

on a.chart\_id = b.chart\_id;

quit;

data final2;

Set final1 (Drop=POS); run;

\*adding in project name & renaming columns;

data final3;

set final2 (rename=(

member\_number=MEMBER

Provider\_Number=PROVIDER

chase\_weight=CHASE\_WEIGHT

provider\_group\_number=PROVIDER\_GROUP

Provider\_First\_Name=PROVIDER\_FIRST\_NAME

Provider\_Last\_Name=PROVIDER\_LAST\_NAME

Provider\_cms\_specialty\_code=CMS\_SPECIALTY\_CODE

chart\_rank=CHASE\_LIST\_RANK

provider\_address\_1=CHART\_ADDRESS1

provider\_address\_2=CHART\_ADDRESS2

provider\_city=CHART\_CITY

provider\_state=CHART\_STATE

provider\_zip=CHART\_ZIP

provider\_phone=PROVIDER\_PHONE

provider\_fax=PROVIDER\_FAX

ccokflag=CCOK\_FLAG

POS\_CODE=POS

));

PROJECT\_NAME= 'CMCROKCOMPI23';

CHART\_IMAGE\_NAME='';

PROVIDER\_EMAIL='';

SPECIAL\_HANDLING\_CATEGORY='';

SPECIAL\_HANDLING\_COMMENT='';

CONTACT\_NAME='';

CONTACT\_PHONE='';

CONTACT\_EXT='';

CONTACT\_FAX='';

USER\_FIELD\_3='';

If CCOK\_FLAG = 0 then CHART\_CHASE\_METHOD = 'Full Service';

else CHART\_CHASE\_METHOD = 'ABS';

If CCOK\_FLAG = 0 then IS\_ABSTRACT = 'N';

else IS\_ABSTRACT = 'Y';

run;

data FINAL4\_20240212152901;

retain CHART\_ID PROJECT\_NAME MEMBER PROVIDER IS\_ABSTRACT CHART\_IMAGE\_NAME

CHASE\_WEIGHT PROVIDER\_GROUP CHASE\_LIST\_RANK CCOK\_FLAG CHART\_CHASE\_METHOD

USER\_FIELD\_3 PROVIDER\_FIRST\_NAME PROVIDER\_LAST\_NAME CMS\_SPECIALTY\_CODE

CHART\_ADDRESS1 CHART\_ADDRESS2 CHART\_CITY CHART\_STATE CHART\_ZIP

PROVIDER\_PHONE PROVIDER\_FAX PROVIDER\_EMAIL POS SPECIAL\_HANDLING\_CATEGORY

SPECIAL\_HANDLING\_COMMENT CONTACT\_NAME CONTACT\_PHONE CONTACT\_EXT CONTACT\_FAX

Member\_Plan\_ID Member\_HIOS\_Product Member\_DOB Member\_Gender Member\_Population

Member\_Rating\_Area Member\_Eligible\_Months Member\_Zip\_Code Member\_Score ;

set final3 (Keep= CHART\_ID PROJECT\_NAME MEMBER PROVIDER IS\_ABSTRACT CHART\_IMAGE\_NAME

CHASE\_WEIGHT PROVIDER\_GROUP CHASE\_LIST\_RANK CCOK\_FLAG CHART\_CHASE\_METHOD

USER\_FIELD\_3 PROVIDER\_FIRST\_NAME PROVIDER\_LAST\_NAME CMS\_SPECIALTY\_CODE

CHART\_ADDRESS1 CHART\_ADDRESS2 CHART\_CITY CHART\_STATE CHART\_ZIP

PROVIDER\_PHONE PROVIDER\_FAX PROVIDER\_EMAIL POS SPECIAL\_HANDLING\_CATEGORY

SPECIAL\_HANDLING\_COMMENT CONTACT\_NAME CONTACT\_PHONE CONTACT\_EXT CONTACT\_FAX

Member\_Plan\_ID Member\_HIOS\_Product Member\_DOB Member\_Gender Member\_Population

Member\_Rating\_Area Member\_Eligible\_Months Member\_Zip\_Code Member\_Score);

if member = 'XXXXXXXXXX' and chase\_list\_rank= 1904 then delete;

if member = 'XXXXXXXXXX' and chase\_list\_rank= 3449 then delete; run;

\*read in final chart chase list submitted to Change;

data final\_list\_submitted;

set final.final4\_20240131153531; Run;

\*import the Client Chase Status report for project CMCROKCOMPI23 from XXXXXXXXXX on 2/12/24;

PROC IMPORT DATAFILE="&outpath\XXXXXXXXXX Chase Status Report 2122024.xlsx"

OUT=work.chase\_status\_report

DBMS=xlsx REPLACE;

SHEET="Sheet1"; /\* Replace Sheet1 with the actual name of the sheet you want to import \*/

GETNAMES=YES; /\* This option tells SAS to use the first row of the Excel sheet as variable names \*/

RUN;

\*Adding variables from chaselist raw to the FINAL4\_20240212152901;

proc sql;

create table chaselist\_combined as

select a.\*, b.Risk\_View\_Member\_ID, b.LOB\_Member\_ID, b.Member\_First\_Name, b.Member\_Last\_Name,

b.Member\_Metal\_Level, b.Member\_PSI, b.Risk\_View\_Provider\_ID, b.Provider\_NPI,

b.Provider\_Tax\_ID, b.Provider\_CMS\_Specialty\_Desc, b.Chart\_Type, b.Is\_PCP, b.Provider\_Group\_Name

from work.FINAL4\_20240212152901 as a

left join Chaselistraw3 as b

on a.chart\_ID = b.chart\_ID;

quit;

\*adding in chart\_ID\_AP from the chase status report from alert portal, joining on combos

of member\_ID and provider NPI;

proc sql;

create table chaselist\_combined1 as

select a.\*, b.chart\_ID as Chart\_ID\_AP

from work.chaselist\_combined as a

left join Chase\_Status\_Report as b

on a.member = b.Client\_Member\_ID and a.Provider = b.client\_provider\_ID;

quit;

\*adding membet social security numbers to combined chaselist;

proc sql;

create table chaselist\_combined2 as

select a.\*, b.SSN as Member\_SSN

from chaselist\_combined1 as a

left join memb.memb as b

on a.member = b.member\_; run;

/\*\*splitting chaselist into just ABS chart chases, where CCOK\_Flag = 1\*/

/\* 2365 observations;\*/

data chaselist\_ABS;

set chaselist\_combined2;

where CCOK\_FLAG=1; run;

\*split list into ones for Christy(560 obs), Saint Fraincis (1805), and Ciox (283 obs - all Full Service);

proc freq data=chaselist\_ABS;

tables Provider\_Tax\_ID ; run;

data chaselist\_Christy;

set chaselist\_ABS;

where provider\_tax\_id in ('XXXXXXXXXX' 'XXXXXXXXXX' 'XXXXXXXXXX'); run;

data chaselist\_Saint\_Francis;

set chaselist\_ABS;

where provider\_tax\_id in ('XXXXXXXXXX' 'XXXXXXXXXX' 'XXXXXXXXXX'); run;

\*0 observations are in the chaselist\_ABS for Ciox tax IDs), checking in chaselist\_combined;

proc sql;

create table chaselist\_XXXXXXXXXX as

select a.\*, b.ccokflag2

from chaselist\_ABS as a

left join orig\_final\_list as b

on a.chart\_ID = b.chart\_ID

where b.ccokflag2 = 1;

quit;

\*creating a review-specific list for Christy then exporting to excel;

data CHASELIST\_XXXXXXXXXX;

Format MEMBER PROVIDER\_FIRST\_NAME PROVIDER\_LAST\_NAME MEMBER\_DOB MEMBER\_GENDER

MEMBER\_FIRST\_NAME MEMBER\_LAST\_NAME CHART\_TYPE CHART\_ID\_AP PROVIDER\_GROUP\_NAME CAPTURED NOTES;

Set CHASELIST\_CHRISTY (Keep=MEMBER PROVIDER\_FIRST\_NAME PROVIDER\_LAST\_NAME MEMBER\_DOB MEMBER\_GENDER

MEMBER\_FIRST\_NAME MEMBER\_LAST\_NAME CHART\_TYPE CHART\_ID\_AP PROVIDER\_GROUP\_NAME);

CAPTURED = '';

NOTES = '';

run;

\*creating a review-specific list of fields for Saint Francis then exporting to Excel;

data CHASELIST\_XXXXXXXXXX;

Format CHART\_ID CHART\_ID\_AP MEMBER MRN MEMBER\_FIRST\_NAME MEMBER\_LAST\_NAME MEMBER\_DOB MEMBER\_GENDER

MEMBER\_SSN;

Set chaselist\_Saint\_Francis (Keep=CHART\_ID CHART\_ID\_AP MEMBER MEMBER\_FIRST\_NAME MEMBER\_LAST\_NAME

MEMBER\_DOB MEMBER\_GENDER MEMBER\_SSN);

MRN= ''; Run;