

```

/*****/

/*
    This program is deployed as weekly job (every Tuesday). See
    NC_REVCYC_NCRHB1223_Weekly_HB_Unbilled_AR_Cubes.sas program in the
    jobflow deployed
    folder.

    The weekly job will obtain the latest report run date, and created
    datasets from ushare:
        1. Create detailed AR unbilled dataset (nonCE)
        2. Create detailed AR unbilled dataset (CE) for report date
        3. Create a summary dataset from the 2 detailed datasets
        4. Append (3) to Unbilled_AR_Cube

    This job can also be run manually to process a particular report run
    date.

    Documentation can be found on the shared drive at:
        finance and reporting\Reporting\AR\Weekly Archiving Process
    Documentation
*/

options sasautos=('/apps/sas/datasets/data26/NCREVCYC/maclib');

/* Get user id and password */

%include
"/apps/sas/datasets/data26/NCREVCYC/code/&sysuserid/userpass.sas";

/* To reference ushare as sas dataset */

libname ushare teradata db=hcclnc_ushare tdpid=tdp1 user=&CLR_NC_USERID
password=&CLR_NC_PW;

/* backup dataset locations */

options compress=yes; /* compressed datasets to save space */

%let detailed =
/apps/sas/datasets/data26/NCREVCYC/data/Unbilled_AR/detailed;
%let summary =
/apps/sas/datasets/data26/NCREVCYC/data/Unbilled_AR/summary;
%let cube = /apps/sas/datasets/data26/NCREVCYC/data/Unbilled_AR/cube;
%let infomap = /apps/sas/datasets/data26/NCREVCYC/data/NC_REVCYC_MAP;

libname detailed "&detailed";
libname summary "&summary";
libname cube "&cube";
libname infomap "&infomap";

```

```

/*
    Step 1:

        The report date to process will be the last date the 1214 Infoview
        job was run.
        We can also override this date with an explicit %let statement.
*/

/* Find maximum report date in ushare */

proc sql;
connect to teradata as hcclnc(db=hcclnc tdpid=tdp1 user=&CLR_NC_USERID
password=&CLR_NC_PW);
create table rundate (compress=yes) as
select * from connection to hcclnc
(
select
    'temp' as temp,
    max(RPT_RUN_DATE) as max_rpt_run_date
from hcclnc_ushare.C405371_UNBILLED_HBA

group by 1
);
disconnect from hcclnc;
quit;

/* Create a macro variable for the run date so that it can be used in the
dataset name */

data _null_;
    set rundate end=lastobs;    /* just incase we have multiple rows */
    if lastobs then call
symput('max_rpt_run_date',left(put(max_rpt_run_date,date9.)));
run;

/*
    If we want to process with an explicit report date, uncomment the
    following %let statement
    and assign the date in date9 format (e.g. 25dec2013)
*/
*%let max_rpt_run_date = 25dec2012;

%put max_rpt_run_date = &max_rpt_run_date;

/*
    Step 2:

        Create backup datasets for detailed non-ce, ce and summary. Also
        create a cube.
*/

```

```

/*
    Create detailed AR unbilled dataset (nonCE) for report run date

    First create from ushare with original table format.  Second from
    ushare with new table format.
    Dates are mutually exclusive between ushare tables.
*/

data detailed.nonce_detail_&max_rpt_run_date;
    set ushare.C405371_UNBILLED_HBA;
        where rpt_run_date = "&max_rpt_run_date"d;
run;

/* Create detailed AR unbilled dataset (CE) for report date */
data detailed.ce_detail_&max_rpt_run_date;
    set ushare.G153063_HBUnbilled_CE;
        where rpt_run_date = "&max_rpt_run_date"d;
run;

/* Create a summary dataset from the 2 detailed datasets */
proc sql;
    create table summary.summary_&max_rpt_run_date as

        select
            RPT_RUN_DATE,
            case
                when (discharge_date is not null and billing_status =
'Discharged/Not Billed') then 'Discharged'
                else 'InHouse'
            end as Discharged_InHouse,
            CODED_HAR,
            ACCT_BASE_CLASS,
            (CASE WHEN LOB IN ('KP Traditional','Medicare Advantage','KP
Medicaid') then 'Capitated Line of Business' else 'Non-capitated Line
of Business' end) as cap,
            lob,
            case
                when (primary_accountability = 'NOT IN ANY WQ' and
hsp_acct_remote_id is not null and pos_primary_a is not null and
(substr(pos_primary_a,1,3) = 'UNK')) then 'UNK'
                when (primary_accountability = 'NOT IN ANY WQ' and
hsp_acct_remote_id is not null and pos_primary_a is not null and
(substr(pos_primary_a,1,3) <> 'UNK')) then pos_primary_a
                when (primary_accountability = 'NOT IN ANY WQ' and
hsp_acct_remote_id is not null and pos_primary_a is null) then 'UNK'
                when (substr(primary_accountability,1,3) = 'UNK') then
'UNK'
                when (primary_accountability is null) then 'UNK'
                else primary_accountability
            end as primary_acct,
            FINANCIAL_CLASS,

```

```

        sum(coalesce(aging_0_30_days,0)) as aging_0_30_days,
        sum(coalesce(aging_31_60_days,0)) as aging_31_60_days,
        sum(coalesce(aging_61_90_days,0)) as aging_61_90_days,

sum(coalesce(aging_31_60_days,0)+coalesce(aging_61_90_days,0)) as
aging_31_90_days,
        sum(coalesce(aging_91_120_days,0)) as aging_91_120_days,
        sum(coalesce(aging_121_150_days,0)) as aging_121_150_days,
        sum(coalesce(aging_151_180_days,0)) as aging_151_180_days,
        sum(coalesce(aging_181_210_days,0)) as aging_181_210_days,
        sum(coalesce(aging_211_240_days,0)) as aging_211_240_days,
        sum(coalesce(aging_241_270_days,0)) as aging_241_270_days,
        sum(coalesce(aging_271_300_days,0)) as aging_271_300_days,
        sum(coalesce(aging_301_330_days,0)) as aging_301_330_days,
        sum(coalesce(aging_331_365_days,0)) as aging_331_365_days,
        sum(coalesce(aging_365_days,0)) as aging_365_days,

sum(coalesce(aging_91_120_days,0)+coalesce(aging_121_150_days,0)+coale
sce(aging_151_180_days,0)+coalesce(aging_181_210_days,0)+

coalesce(aging_211_240_days,0)+coalesce(aging_241_270_days,0)+coalesce
(aging_271_300_days,0)+coalesce(aging_301_330_days,0)+

coalesce(aging_331_365_days)+coalesce(aging_365_days,0)) as
aging_91_365plus_days,
        sum(coalesce(tot_acct_bal,0)) as tot_acct_bal,
        sum(coalesce(total_charges,0)) as total_charges,
        count(*) as cases
from detailed.noncase_detail_&max_rpt_run_date
group by 1,2,3,4,5,6,7,8

        union all

select
RPT_RUN_DATE,
case
        when (discharge_date is not null and billing_status =
'Billed') then 'Discharged'
        else 'InHouse'
end as Discharged_InHouse,
CODED_HAR,
ACCT_BASE_CLASS,
(CASE WHEN LOB IN ('KP Traditional','Medicare Advantage','KP
Medicaid') then 'Capitated Line of Business' else 'Non-capitated Line
of Business' end) as cap,
lob,
primary_accountability as primary_acct,
FINANCIAL_CLASS,
sum(coalesce(aging_0_30_days,0)) as aging_0_30_days,
sum(coalesce(aging_31_60_days,0)) as aging_31_60_days,
sum(coalesce(aging_61_90_days,0)) as aging_61_90_days,

```

```

sum(coalesce(aging_31_60_days,0)+coalesce(aging_61_90_days,0)) as
aging_31_90_days,
    sum(coalesce(aging_91_120_days,0)) as aging_91_120_days,
    sum(coalesce(aging_121_150_days,0)) as aging_121_150_days,
    sum(coalesce(aging_151_180_days,0)) as aging_151_180_days,
    sum(coalesce(aging_181_210_days,0)) as aging_181_210_days,
    sum(coalesce(aging_211_240_days,0)) as aging_211_240_days,
    sum(coalesce(aging_241_270_days,0)) as aging_241_270_days,
    sum(coalesce(aging_271_300_days,0)) as aging_271_300_days,
    sum(coalesce(aging_301_330_days,0)) as aging_301_330_days,
    sum(coalesce(aging_331_365_days,0)) as aging_331_365_days,
    sum(coalesce(aging_365_days,0)) as aging_365_days,

sum(coalesce(aging_91_120_days,0)+coalesce(aging_121_150_days,0)+coale
sce(aging_151_180_days,0)+coalesce(aging_181_210_days,0)+

coalesce(aging_211_240_days,0)+coalesce(aging_241_270_days,0)+coalesce
(aging_271_300_days,0)+coalesce(aging_301_330_days,0)+

coalesce(aging_331_365_days,0)+coalesce(aging_365_days,0)) as
aging_91_365plus_days,
    sum(coalesce(tot_acct_bal)) as tot_acct_bal,
    sum(coalesce(total_charges)) as total_charges,
    count(*) as cases
from detailed.ce_detail &max_rpt_run_date
    group by 1,2,3,4,5,6,7,8;
quit;

/*
    Step 3:

        Create cube.
*/

/*
    Append latest summary to current cube to produce a dated cube

    The where clause is necessary to make sure we don't append the same
    data multiple times if we run the program multiple time.
*/
data cube.unbilled_ar_cube &max_rpt_run_date;
    set cube.unbilled_ar_cube_current(where=(rpt_run_date ne
"&max_rpt_run_date"d)
        summary.summary_&max_rpt_run_date;
run;

/* Reset the current cube--will be used for reporting */
data cube.unbilled_ar_cube_current;
    set cube.unbilled_ar_cube &max_rpt_run_date;
run;

```

/*

Step 4:

Create detailed datasets for Infomap that can be feed into Web Report Studio.

The 1216 and 1217 queries add the following columns:

Discharged_InHouse

cap

primary_acct

Reproduce that logic here so the resulting datasets can be used to produce the 1216 and 1217 results from within Web Report Studio.

Also create a Infomap cube dataset.

*/

proc sql;

```
    create table infomap.nonce_detail_current as
    select
        detail.*,
    case
        when (discharge_date is not null and billing_status =
'Discharged/Not Billed') then 'Discharged'
        else 'InHouse'
    end as Discharged_InHouse,
    (CASE WHEN LOB IN ('KP Traditional','Medicare Advantage','KP
Medicaid') then 'Capitated Line of Business' else 'Non-capitated Line
of Business' end) as cap,
    case
        when (primary_accountability = 'NOT IN ANY WQ' and
hsp_acct_remote_id is not null and pos_primary_a is not null and
(substr(pos_primary_a,1,3) = 'UNK')) then 'UNK'
        when (primary_accountability = 'NOT IN ANY WQ' and
hsp_acct_remote_id is not null and pos_primary_a is not null and
(substr(pos_primary_a,1,3) <> 'UNK')) then pos_primary_a
        when (primary_accountability = 'NOT IN ANY WQ' and
hsp_acct_remote_id is not null and pos_primary_a is null) then 'UNK'
        when (substr(primary_accountability,1,3) = 'UNK') then
'UNK'
        when (primary_accountability is null) then 'UNK'
        else primary_accountability
    end as primary_acct
    from detailed.nonce_detail_&max_rpt_run_date detail;

    create table infomap.ce_detail_current as
    select
        detail.*,
    case
        when (discharge_date is not null and billing_status =
'Discharged/Not Billed') then 'Discharged'
        else 'InHouse'
```

```

        end as Discharged_InHouse,
        (CASE WHEN LOB IN ('KP Traditional','Medicare Advantage','KP
Medicaid') then 'Capitated Line of Business' else 'Non-capitated Line
of Business' end) as cap,
        primary_accountability as primary_acct
        from detailed.ce_detail_&max_rpt_run_date detail;

        create table infomap.unbilled_ar_cube_current as
select *
        from cube.unbilled_ar_cube_current;

quit;

/* Produce a csv file for e-mail */
%let attachment="&cube/unbilled_ar_cube_&max_rpt_run_date..csv";
proc export data=cube.unbilled_ar_cube_&max_rpt_run_date
        outfile=&attachment
        dbms=csv replace;
run;

filename outmail email
/*          to="finance.and.reporting@kp.org"          */
        to="stephen.x.fiorelli@kp.org"
        from="finance.and.reporting@kp.org"
        subject="(Auto e-mail): (phi) NCRHB1223 Weekly HB Unbilled AR
Cube"
        attach=(&attachment. CT='BIN' name
="Weekly_HB_Unbilled_AR_Cube" EXT='CSV');

data _null_;
        file outmail ;
        put "Please find attached file.";
run;

/*
        Step 4:

                Make sure permissions on all datasets created is group writeable
and remove
                world read (o-r).
*/

%chmod(dir=&detailed,permissions=g+w);
%chmod(dir=&detailed,permissions=o-r);

%chmod(dir=&summary,permissions=g+w);
%chmod(dir=&summary,permissions=o-r);

%chmod(dir=&cube,permissions=g+w);
%chmod(dir=&cube,permissions=o-r);

```

