**CREATE** **OR** **REPLACE** **PACKAGE** **BODY** sweeps

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

*sweeps*

*<pre>*

*Artisan Date Comments*

*============ ========= ========================================================*

*Lobo-Blanco 2009Feb01 RECORDS-1179 Created for sweep feature.*

*bcoulam 2009Feb27 RECORDS-1179 Rewrite.*

*bcoulam 2009Mar20 RECORDS-1029 Adding sweep tally code.*

*bcoulam 2009Jun12 RECORDS-2138 Rewrite.*

*bcoulam 2009Aug03 RECORDS-3702 Mucho modifications.*

*bcoulam 2010Dec13 RECORDS-4590 Removed rspbl\_org\_sens\_flg column.*

*</pre>*

*Copyright 2009 Intellectual Reserve, Inc.*

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

**AS**

*--------------------------------------------------------------------------------*

*-- PACKAGE CONSTANTS, VARIABLES, TYPES, EXCEPTIONS*

*--------------------------------------------------------------------------------*

gc\_start **CONSTANT** **VARCHAR2**(10) := 'START';

gc\_update **CONSTANT** **VARCHAR2**(10) := 'UPDATE';

gc\_end **CONSTANT** **VARCHAR2**(10) := 'END';

*--------------------------------------------------------------------------------*

*-- PRIVATE FUNCTIONS AND PROCEDURES*

*--------------------------------------------------------------------------------*

*/\*\*-----------------------------------------------------------------------------*

*load\_sweep\_set*

*Fills the temporary table with IDs of those missionaries that need to be picked*

*up for either sweep or tally purposes. Since the tally needs to count prefield*

*missionaries, we end up including a number of prefield missionaries who probably*

*aren't needed in the sweep, but that is how legacy did it, which we were told*

*to replicate.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** load\_sweep\_set(i\_sweep\_dt **IN** **DATE**)

**IS**

**BEGIN**

**DELETE** **FROM** fin\_sweep\_msny\_gt;

**INSERT** **INTO** fin\_sweep\_msny\_gt

**SELECT** m.id

,m.enamem\_id

,m.legacy\_miss\_id

,m.mrn

,mn.first\_nm||**DECODE**(em.misstyp\_id,3,' and '||mnw.first\_nm,**NULL**)

,mn.last\_nm

,**DECODE**(em.misstyp\_id,3,**NULL**,mn.middle\_nm)

,em.term\_months pmts\_to\_send

,**SUM**(**DECODE**(fph.dr\_cr\_code, 'DR', 1, 'CR', -1)) pmts\_sent

**FROM** enabled\_members em

**JOIN** missionaries m **ON** m.legacy\_miss\_id = em.legacy\_miss\_id *-- only joins husband for couples*

**JOIN** current\_perm\_assignment\_vw cpa **ON** cpa.call\_emem\_id = em.enamem\_id **AND** cpa.early\_release\_flag <> 'Y' *-- RMH Records-4102 Filter out early released missionaries*

**JOIN** msny\_name mn **ON** mn.legacy\_miss\_id = em.legacy\_miss\_id **AND** mn.name\_type\_id = 10 *-- Candidate name*

**LEFT** **JOIN** msny\_name mnw **ON** mnw.legacy\_miss\_id = em.legacy\_spouse\_miss\_id **AND** mnw.name\_type\_id = 10 *-- Wife's Candidate name*

**LEFT** **JOIN** fin\_pay\_hist fph **ON** fph.legacy\_miss\_id = m.legacy\_miss\_id

**WHERE** em.procstat\_id **IN** (11, 12, 13, 14)

**AND** em.enamem\_release\_date >= i\_sweep\_dt

**GROUP** **BY** m.id

,m.enamem\_id

,m.legacy\_miss\_id

,m.mrn

,mn.first\_nm||**DECODE**(em.misstyp\_id,3,' and '||mnw.first\_nm,**NULL**)

,mn.last\_nm

,**DECODE**(em.misstyp\_id,3,**NULL**,mn.middle\_nm)

,em.term\_months;

*-- This was the new filter, rejected in favor of the Legacy filter used above.*

*-- The legacy filter currently (July 2009) includes about 5000+ more missionaries*

*-- than the new filter would have.*

*-- (*

*-- -- 12=Started, 13=Arrived, 43=On leave from MTC, 44= On leave from Field*

*-- em.procstat\_id IN (12, 13, 43, 44)*

*-- OR*

*-- -- 11=Assigned and eff dt of sweep is same as month of mtc start*

*-- (em.procstat\_id = 11 AND TRUNC(em.enamem\_mission\_start\_date, 'Month') = i\_sweep\_dt)*

*-- OR*

*-- -- 14=Released, but date of sweep is same as month of release*

*-- (em.procstat\_id = 14 AND TRUNC(em.enamem\_release\_date, 'Month') = i\_sweep\_dt)*

*-- )*

**END** load\_sweep\_set;

*/\*\*-----------------------------------------------------------------------------*

*sweep\_finalized*

*Returns TRUE if the last run for the given sweep month was already Finalized,*

*else it returns FALSE. This allows users to attempt the sweep multiple times,*

*but only one capped off as FINALIZED will prevent further sweep runs for the*

*same month.*

*------------------------------------------------------------------------------\*/*

**FUNCTION** sweep\_finalized(i\_sweep\_dt **IN** **DATE**)

**RETURN** **BOOLEAN**

**IS**

l\_finalized fin\_proc\_hist.status\_cd%**TYPE**;

**BEGIN**

*-- check for the existence of the Finalized tag.*

**SELECT** status\_cd

**INTO** l\_finalized

**FROM** fin\_proc\_hist

**WHERE** proc\_hist\_id = (**SELECT** **MAX**(proc\_hist\_id)

**FROM** fin\_proc\_hist

**WHERE** proc\_nm = sweep\_process\_nm

**AND** proc\_dt = i\_sweep\_dt);

**IF** (**INSTR**(l\_finalized, 'FINALIZED') > 0) **THEN**

**RETURN** **TRUE**;

**ELSE**

**RETURN** **FALSE**;

**END** **IF**;

**EXCEPTION**

**WHEN** **NO\_DATA\_FOUND** **THEN**

**RETURN** **FALSE**;

**END** sweep\_finalized;

*/\*\*-----------------------------------------------------------------------------*

*tag\_proc\_hist*

*Used to note the start, progress and finish of a sweep run. If there are unexpected*

*errors the end\_dt will be empty and the process text field will still indicate*

*it has only started.*

*%param i\_sweep\_dt Date of the sweeps being tracked in fin\_proc\_hist.*

*%param i\_op Whether or not this tagging operation is due to a process starting*

*(use global constant gc\_INSERT), noting a recent state change (use*

*constant gc\_UPDATE), or a process ending (use gc\_END).*

*%param io\_proc\_hist\_id For the insert operation, captures the new sequence-generated*

*surrogate key ID and returns it to the caller to be used*

*in the update/finish call when the process is done.*

*%param i\_tag\_txt Text to use in "tagging" the row being inserted to fin\_proc\_hist*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** tag\_proc\_hist

(

i\_sweep\_dt **IN** **DATE**,

i\_op **IN** **VARCHAR2**,

io\_proc\_hist\_id **IN** **OUT** fin\_proc\_hist.proc\_hist\_id%**TYPE**,

i\_status\_cd **IN** fin\_proc\_hist.status\_cd%**TYPE** **DEFAULT** **NULL**,

i\_tag\_txt **IN** fin\_proc\_hist.proc\_txt%**TYPE** **DEFAULT** **NULL**

) **IS**

**PRAGMA** **AUTONOMOUS\_TRANSACTION**;

**BEGIN**

logs.dbg(i\_tag\_txt);

**IF** (i\_op = gc\_start) **THEN**

**INSERT** **INTO** fin\_proc\_hist

(proc\_hist\_id

,proc\_nm

,proc\_dt

,run\_dt

,end\_dt

,status\_cd

,proc\_txt

,client\_id)

**VALUES**

(fin\_proc\_hist\_seq.NEXTVAL

,sweep\_process\_nm

,i\_sweep\_dt

,**SYSDATE**

,**NULL**

,**NVL**(i\_status\_cd,'RUNNING')

,i\_tag\_txt

,env.get\_client\_id)

**RETURNING** proc\_hist\_id **INTO** io\_proc\_hist\_id;

**ELSIF** (i\_op = gc\_update) **THEN**

**UPDATE** fin\_proc\_hist

**SET** status\_cd = **NVL**(i\_status\_cd,status\_cd)

,proc\_txt = **NVL**(i\_tag\_txt,'Updated')

**WHERE** proc\_hist\_id = io\_proc\_hist\_id;

**ELSIF** (i\_op = gc\_end) **THEN**

**UPDATE** fin\_proc\_hist

**SET** end\_dt = **SYSDATE**

,status\_cd = **NVL**(i\_status\_cd, 'SUCCESS')

,proc\_txt = **CASE** **WHEN** i\_tag\_txt **IS** **NULL** **THEN** 'Ended'

**WHEN** i\_status\_cd **IN** ('FAILED','FAILURE','ERROR') **THEN**

*-- preserve the last action with the failure text*

'Last action: ' || proc\_txt || **CHR**(10)||

'Error msg: '||i\_tag\_txt

**ELSE** i\_tag\_txt

**END**

,client\_id = env.get\_client\_id

**WHERE** proc\_hist\_id = io\_proc\_hist\_id;

**END** **IF**;

**COMMIT**;

**END** tag\_proc\_hist;

*/\*\*-----------------------------------------------------------------------------*

*clean\_prior\_payments*

*Remove any rows for the given sweep date. We do not delete manual adjustment*

*rows on purpose. Users typically do not make adjustments until after the current*

*month's sweep has been finalized. Once finalized, this routine will never be*

*called for that month. The users can also, in theory, enter debits or credits*

*for future dates. We also do not want to delete those adjustments. So we only*

*delete adjustments made by the backend PL/SQL.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** clean\_prior\_payments(i\_sweep\_dt **IN** **DATE**)

**IS**

**BEGIN**

logs.dbg('Clean out payment history');

**DELETE** **FROM** fin\_pay\_hist

**WHERE** pay\_dt = i\_sweep\_dt

**AND** mod\_by = 'SYSTEM';

**END** clean\_prior\_payments;

*/\*\*-----------------------------------------------------------------------------*

*clean\_prior\_sweep*

*Remove any data for the given sweep date.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** clean\_prior\_sweep(i\_sweep\_dt **IN** **DATE**)

**IS**

**BEGIN**

clean\_prior\_payments(i\_sweep\_dt);

logs.dbg('Clean out sweep tally');

**DELETE** **FROM** fin\_sweep\_tally

**WHERE** sweep\_dt = i\_sweep\_dt;

logs.dbg('Clean out sweep missionary');

**DELETE** **FROM** fin\_sweep\_msny

**WHERE** sweep\_dt = i\_sweep\_dt;

**END** clean\_prior\_sweep;

*/\*\*-----------------------------------------------------------------------------*

*adj\_msny\_fund\_type\_id*

*Finds new changes to fund\_type\_id (can be overridden for unit in fin\_unit\_override*

*or changed for country on fin\_poldivs) and applies them to the missionaries they*

*affect.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** adj\_msny\_fund\_type\_id

**IS**

**TYPE** tr\_fund\_type\_chg **IS** **RECORD** (

enamem\_id **INTEGER**

, old\_fund\_type\_id **INTEGER**

, new\_fund\_type\_id **INTEGER**

);

**TYPE** tnt\_changes **IS** **TABLE** **OF** tr\_fund\_type\_chg **INDEX** **BY** **PLS\_INTEGER**;

lnt\_changes tnt\_changes;

**BEGIN**

**SELECT** enamem\_id,

old\_fund\_type\_id,

new\_fund\_type\_id

**BULK** **COLLECT** **INTO** lnt\_changes

**FROM** (

**SELECT** */\*+ DYNAMIC\_SAMPLING(fmsg 2) \*/*

fmf.enamem\_id

*-- KEEP THESE COMMENTED-OUT COLUMNS AROUND FOR DEBUGGING*

*--,fsmg.legacy\_miss\_id*

*--,em.misstyp\_id*

*--,em.procstat\_id*

*--,DECODE(uo.unit\_override\_id,NULL,'N','Y') override*

*--,fmf.fund\_unit\_id*

*--,com.orglongname AS fund\_unit\_nm*

*--,ptc.primary\_longname AS fund\_unit\_loc*

*--,ptc.country\_longname AS fund\_unit\_cntry*

,fmf.fund\_type\_id **AS** old\_fund\_type\_id

*-- if unit not overriden, and not a US territory, use the country's funding method,*

*-- otherwise use the funding method of the US territory or override record*

,**NVL**(uo.fund\_type\_id, **NVL**(ust.fund\_type\_id,fp.fund\_type\_id)) **AS** new\_fund\_type\_id

**FROM** fin\_sweep\_msny\_gt fsmg *-- limit changes to sweep set*

**JOIN** fin\_msny\_funding fmf **ON** fmf.enamem\_id = fsmg.enamem\_id

**JOIN** enabled\_members em **ON** em.enamem\_id = fmf.enamem\_id

**JOIN** fin\_organization\_mv com **ON** com.orgnbr = fmf.fund\_unit\_id

**LEFT** **JOIN** poldiv\_to\_country\_mv ptc **ON** ptc.primary\_poldivid = com.primarylocpoldivid

*-- join to FUO to get unit's funding method overriding country setting*

**LEFT** **JOIN** fin\_unit\_override uo **ON** uo.orgnbr = **CASE**

**WHEN** fmf.allow\_override\_yn = 'Y' **THEN**

com.orgnbr

**ELSE**

**NULL** *-- prevent join if missionary is "locked" from unit overriding*

**END**

*-- join to FP on country to get country's funding method*

**LEFT** **JOIN** fin\_poldivs fp **ON** fp.poldivid = ptc.country\_poldivid

*-- join inline view in case it is a US territory that must remain GMF*

**LEFT** **JOIN** (

**SELECT** cpm.poldivid

*--,cpm.longname*

,**NVL**(fp.fund\_type\_id, 2) **AS** fund\_type\_id

**FROM** col\_poldiv\_mv cpm

**LEFT** **JOIN** fin\_poldivs fp **ON** fp.poldivid = cpm.poldivid

**WHERE** cpm.parpoldivid = 799

**AND** poltypedesctxt **IN** ('Territory', 'Country')

) ust **ON** ust.poldivid = ptc.primary\_poldivid

**WHERE** fmf.fund\_unit\_id **IS** **NOT** **NULL** *-- all juniors should have funding unit*

**AND** em.misstyp\_id **NOT** **IN** (3,4,5)

)

**WHERE** new\_fund\_type\_id **IS** **NOT** **NULL** *-- don't take null as a valid new fund type id*

**AND** old\_fund\_type\_id <> new\_fund\_type\_id;

**IF** (lnt\_changes **IS** **NOT** **NULL** **AND** lnt\_changes.COUNT > 0) **THEN**

**FOR** i **IN** lnt\_changes.FIRST..lnt\_changes.LAST **LOOP**

**UPDATE** fin\_msny\_funding

**SET** fund\_type\_id = lnt\_changes(i).new\_fund\_type\_id

**WHERE** enamem\_id = lnt\_changes(i).enamem\_id;

**END** **LOOP**;

**END** **IF**;

logs.dbg('Updated fund\_type\_id for '||**SQL**%**ROWCOUNT**||' enabled members.');

**END** adj\_msny\_fund\_type\_id;

*/\*\*-----------------------------------------------------------------------------*

*ins\_sweep\_msny*

*Captures missionaries active this month and their funding metadata.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** ins\_sweep\_msny(i\_sweep\_dt **IN** **DATE**)

**IS**

l\_created\_dt **DATE** := **SYSDATE**;

**BEGIN**

**INSERT** **INTO** fin\_sweep\_msny

(

sweep\_dt

,created\_dt

,msny\_id

,legacy\_miss\_id

,enamem\_id

,procstat\_id

,msny\_nm

,misstyp\_id

,term\_months

,live\_at\_home\_yn

,mtc\_dt

,arriv\_dt

,release\_dt

,anniv\_dt

,mtc\_id

,mtc\_cd

,mtc\_nm

,call\_type\_id

,comp\_id

,comp\_nm

,comp\_asgloc\_orgnbr

,comp\_asgloc\_id

,comp\_asgloc\_nm

,comp\_asgloc\_orgtype\_cd

,comp\_asgloc\_orgtype\_nm

,comp\_asgloc\_country\_id

,comp\_asgloc\_country\_nm

,comp\_asgloc\_sens\_flg

,asgloc\_orgnbr

,asgloc\_id

,asgloc\_nm

,asgloc\_orgtype\_cd

,asgloc\_orgtype\_nm

,asgloc\_area\_orgnbr

,asgloc\_area\_nm

,asgloc\_country\_id

,asgloc\_country\_nm

,asgloc\_sens\_flg

,within\_msn\_orgnbr

,within\_msn\_asgloc\_id

,within\_msn\_nm

,within\_msn\_sens\_flg

,eccl\_org\_orgnbr

,eccl\_org\_nm

,eccl\_org\_sens\_flg

,temp\_comp\_id

,temp\_comp\_nm

,temp\_comp\_asgloc\_orgnbr

,temp\_comp\_asgloc\_id

,temp\_comp\_asgloc\_nm

,temp\_comp\_asgloc\_sens\_flg

,temp\_asgloc\_orgnbr

,temp\_asgloc\_id

,temp\_asgloc\_nm

,temp\_asgloc\_sens\_flg

,temp\_within\_msn\_orgnbr

,temp\_within\_msn\_asgloc\_id

,temp\_within\_msn\_nm

,temp\_within\_msn\_sens\_flg

,temp\_eccl\_org\_orgnbr

,temp\_eccl\_org\_nm

,temp\_eccl\_org\_sens\_flg

,fund\_type\_id

,last\_pmt\_dt

,blbl\_month\_yn

,crnt\_pmt\_num

,total\_pmts

,fund\_unit\_id

,fund\_unit\_nm

,fund\_unit\_sens\_flg

,fund\_unit\_stake\_id

,fund\_unit\_stake\_nm

,fund\_unit\_stake\_sens\_flg

,fund\_unit\_area\_orgnbr

,fund\_unit\_area\_nm

,fund\_unit\_country\_id

,fund\_unit\_country\_nm

,fund\_unit\_currency\_used

,fund\_unit\_currency\_cd

,fund\_unit\_currency\_nm

,fund\_unit\_exchange\_rt

,fund\_unit\_equal\_amt

,fund\_unit\_commit\_amt

,fund\_unit\_local\_amt

,fund\_unit\_us\_amt

,msn\_currency\_used

,msn\_currency\_cd

,msn\_currency\_nm

,msn\_local\_amt

,msn\_us\_amt

,msn\_exchange\_rt

,msn\_facs\_num

)

**SELECT** i\_sweep\_dt **AS** sweep\_dt

,l\_created\_dt **AS** created\_dt

,gt.msny\_id

,gt.legacy\_miss\_id

,gt.enamem\_id

,em.procstat\_id

,**RTRIM**(gt.last\_nm || ', ' || gt.first\_nm || ' ' || gt.middle\_nm) **AS** msny\_nm

,em.misstyp\_id

,em.term\_months

,**DECODE**(alc.comp\_housing\_id,3,'Y','N') **AS** live\_at\_home\_yn *-- fixed during Sr. Exp work*

,em.enamem\_mission\_start\_date **AS** mtc\_dt

,ca.asgmnt\_start\_date **AS** arriv\_dt

,em.enamem\_release\_date **AS** release\_dt

,em.enamem\_anniversary\_date **AS** anniv\_dt

,ca.call\_mtcs\_id **AS** mtc\_id

,ctom.orgshortname **AS** mtc\_cd

,tf.name **AS** mtc\_nm

,ca.call\_type **AS** call\_type\_id

,alc.comp\_id **AS** comp\_id

,alc.comp\_description **AS** comp\_nm

,cfom.orgnbr **AS** comp\_asgloc\_orgnbr

,alc.asgloc\_id **AS** comp\_asgloc\_id

,cal.name **AS** comp\_asgloc\_nm

,cfom.orgtypecode **AS** comp\_asgloc\_orgtype\_cd

,cfom.orgtypeshortdesctxt **AS** comp\_asgloc\_orgtype\_nm

,cpc.country\_poldivid **AS** comp\_asgloc\_country\_id

,cpc.country\_longname **AS** comp\_asgloc\_country\_nm

,cfom.orgsensitivitycode **AS** comp\_asgloc\_sens\_flg

,mfom.orgnbr **AS** asgloc\_orgnbr

,ca.call\_aloc\_id **AS** asgloc\_id

,mama.asgloc\_name **AS** asgloc\_nm

,mfom.orgtypecode **AS** asgloc\_orgtype\_cd

,mfom.orgtypeshortdesctxt **AS** asgloc\_orgtype\_nm

,mama.area\_orgnbr **AS** asgloc\_area\_orgnbr

,mama.area\_name **AS** asgloc\_area\_nm

,mpc.country\_poldivid **AS** asgloc\_country\_id

,mpc.country\_longname **AS** asgloc\_country\_nm

,mfom.orgsensitivitycode **AS** asgloc\_sens\_flg

,mama.wm\_orgnbr **AS** within\_msn\_orgnbr

,wmfom.asgloc\_id **AS** within\_msn\_asgloc\_id

,mama.wm\_orglongname **AS** within\_msn\_nm

,wmfom.orgsensitivitycode **AS** within\_msn\_sens\_flg

,mama.eccl\_orgnbr **AS** eccl\_org\_orgnbr

,mama.eccl\_orglongname **AS** eccl\_org\_nm

,efom.orgsensitivitycode **AS** eccl\_org\_sens\_flg

,cta.call\_comp\_id **as** temp\_comp\_id

,talc.comp\_description **AS** temp\_comp\_nm

,tcfom.orgnbr **AS** temp\_comp\_asgloc\_orgnbr

,talc.asgloc\_id **AS** temp\_comp\_asgloc\_id

,tcal.name **AS** temp\_comp\_asgloc\_nm

,**NVL**(tcfom.orgsensitivitycode,0) **AS** temp\_comp\_asgloc\_sens\_flg

,tmfom.orgnbr **AS** temp\_asgloc\_orgnbr

,cta.call\_aloc\_id **AS** temp\_asgloc\_id

,tama.asgloc\_name **AS** temp\_asgloc\_nm

,**NVL**(tmfom.orgsensitivitycode,0) **AS** temp\_asgloc\_sens\_flg

,tama.wm\_orgnbr **AS** temp\_within\_msn\_orgnbr

,twmfom.asgloc\_id **AS** temp\_within\_msn\_asgloc\_id

,tama.wm\_orglongname **AS** temp\_within\_msn\_nm

,**NVL**(twmfom.orgsensitivitycode,0) **AS** temp\_within\_msn\_sens\_flg

,tama.eccl\_orgnbr **AS** temp\_eccl\_org\_orgnbr

,tama.eccl\_orglongname **AS** temp\_eccl\_org\_nm

,**NVL**(tefom.orgsensitivitycode,0) **AS** temp\_eccl\_org\_sens\_flg

,fmf.fund\_type\_id

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** lpm.last\_pay\_dt **ELSE** **NULL** **END** **AS** last\_pmt\_dt

,**CASE**

**WHEN** (lpm.last\_pay\_dt **is** **NULL** **OR** fmf.fund\_type\_id **IS** **NULL**)

**THEN** 'N' *-- not enough to go on*

**WHEN** (em.procstat\_id = 11 **AND** **TRUNC**(em.enamem\_mission\_start\_date, 'Month') > i\_sweep\_dt)

**THEN** 'N' *-- prefield that won't be in the MTC this month*

**WHEN** fmf.fund\_type\_id = 3 **OR** (**MONTHS\_BETWEEN**(**TRUNC**(lpm.last\_pay\_dt,'Month'),i\_sweep\_dt) < 0)

**THEN** 'N' *-- unsupported or those whose last payment month has passed*

**ELSE** 'Y'

**END** **AS** blbl\_month\_yn

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** **NVL**(gt.pmts\_sent,0)+1 **ELSE** **NULL** **END** **AS** crnt\_pmt\_num

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** **NVL**(gt.pmts\_to\_send,0) **ELSE** **NULL** **END** **AS** total\_pmts

,fmf.fund\_unit\_id **AS** fund\_unit\_id

,ffom.orglongname **AS** fund\_unit\_nm

,**NVL**(ffom.orgsensitivitycode,0) **AS** fund\_unit\_sens\_flg

,ffom.parentorgnbr **AS** fund\_unit\_stake\_orgnbr

,fstake.orglongname **AS** fund\_unit\_stake\_nm

,**NVL**(fstake.orgsensitivitycode,0) **AS** fund\_unit\_stake\_sens\_flg

,ffom.areaorgnbr **AS** fund\_unit\_area\_orgnbr

,farea.orglongname **AS** fund\_unit\_area\_nm

,ffpc.country\_poldivid **AS** fund\_unit\_country\_id

,ffpc.country\_longname **AS** fund\_unit\_country\_nm

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** ffp.cfis\_local\_or\_us\_flag **ELSE** **NULL** **END** **AS** fund\_unit\_currency\_used

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** fcurr.iso\_currency\_code **ELSE** **NULL** **END** **AS** fund\_unit\_currency\_cd

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** fcurr.currency\_name **ELSE** **NULL** **END** **AS** fund\_unit\_currency\_nm

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** fpcr.exch\_rate **ELSE** **NULL** **END** **AS** fund\_unit\_exchange\_rt

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN** ffp.finpol\_equalized\_pay\_amt

**WHEN** fmf.fund\_type\_id **IN** (4,5) **THEN** alc.housing\_exp

**ELSE** **NULL**

**END** **AS** fund\_unit\_equal\_amt

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN**

*-- add up commitment sources as reported by junior missionary*

**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0)

**WHEN** fmf.fund\_type\_id **IN** (4,5) **THEN**

*-- get housing commitment amount*

fmf.housing\_cmt\_amt

**END** **AS** fund\_unit\_commit\_amt

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN**

*-- Local Amount Due from Funding Unit = commitments if not equalized, otherwise the equalized amount*

*-- This formula for local and US amounts came from Legacy Pro\*C file sweeptally.pc*

**CASE**

*-- check unit override table for equalization flag, before using eq flag for country*

**WHEN** **NVL**(fuo.equalized\_yn, ffp.finpol\_equalized\_yn) = 'Y' **THEN**

ffp.finpol\_equalized\_pay\_amt

**ELSE**

**LEAST**(**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0),

**CASE**

**WHEN** (ffp.finpol\_equalized\_pay\_amt **IS** **NULL** **OR** ffp.finpol\_equalized\_pay\_amt = 0) **THEN**

**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0)

**ELSE**

ffp.finpol\_equalized\_pay\_amt

**END**

)

**END**

**WHEN** fmf.fund\_type\_id = 4 **THEN**

**CASE** **WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y' **AND** fmf.housing\_exp\_amt > 0) **THEN**

fmf.housing\_exp\_amt *-- missionary housing expense is stored in local currency of funding unit*

**ELSE** 0

**END**

**WHEN** fmf.fund\_type\_id = 5 **THEN**

**CASE**

**WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y') **THEN**

*-- Compare housing commitment to housing expense. Take the lesser.*

**CASE** **WHEN** fmf.housing\_cmt\_amt <= fmf.housing\_exp\_amt **THEN**

fmf.housing\_cmt\_amt

**WHEN** fmf.housing\_cmt\_amt > fmf.housing\_exp\_amt **THEN**

fmf.housing\_exp\_amt

**END**

**ELSE** 0

**END**

**ELSE** 0

**END** **AS** fund\_unit\_local\_amt

*-- USD amt due same as local, but multiplied by USD exchange rate*

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN**

*-- USD Amount Due from Funding Unit = commitments if not equalized, otherwise the equalized amount*

*-- This formula for local and US amounts came from Legacy Pro\*C file sweeptally.pc*

**ROUND**(

**CASE**

*-- check unit override table for equalization flag, before using eq flag for country*

**WHEN** **NVL**(fuo.equalized\_yn, ffp.finpol\_equalized\_yn) = 'Y' **THEN**

ffp.finpol\_equalized\_pay\_amt

**ELSE**

**LEAST**(**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0),

**CASE**

**WHEN** (ffp.finpol\_equalized\_pay\_amt **IS** **NULL** **OR** ffp.finpol\_equalized\_pay\_amt = 0) **THEN**

**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0)

**ELSE**

ffp.finpol\_equalized\_pay\_amt

**END**

)

**END** \* fpcr.exch\_rate

)

**WHEN** fmf.fund\_type\_id = 4 **THEN**

**CASE** **WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y' **AND** fmf.housing\_exp\_amt > 0) **THEN**

**ROUND**(fmf.housing\_exp\_amt \* fpcr.exch\_rate)*-- missionary housing expense in terms of USD*

**ELSE** 0

**END**

**WHEN** fmf.fund\_type\_id = 5 **THEN**

**CASE**

**WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y') **THEN**

*-- Compare commitment to housing expense and take the lesser*

**ROUND**(

**CASE** **WHEN** fmf.housing\_cmt\_amt <= fmf.housing\_exp\_amt **THEN**

fmf.housing\_cmt\_amt

**WHEN** fmf.housing\_cmt\_amt > fmf.housing\_exp\_amt **THEN**

fmf.housing\_exp\_amt

**END** \* fpcr.exch\_rate

)

**ELSE** 0

**END**

**ELSE** 0

**END** **AS** fund\_unit\_us\_amt

,mfp.cfis\_local\_or\_us\_flag **AS** msn\_currency\_used

,mcurr.iso\_currency\_code **AS** msn\_currency\_cd

,mcurr.currency\_name **AS** msn\_currency\_nm

*-- this formula for local and US amounts came from Legacy Pro\*C file sweeptally.pc*

,**CASE** **WHEN** mfp.cfis\_local\_or\_us\_flag = 'L' **THEN** mfa.base\_amt **ELSE** **ROUND**(mfa.base\_amt / mpcr.exch\_rate) **END** **AS** msn\_local\_amt

,**CASE** **WHEN** mfp.cfis\_local\_or\_us\_flag = 'L' **THEN** **ROUND**(mfa.base\_amt \* mpcr.exch\_rate) **ELSE** mfa.base\_amt **END** **AS** msn\_us\_amt

,mpcr.exch\_rate **AS** msn\_exchange\_rt

,mfa.gl\_unit\_num **AS** msn\_facs\_num

**FROM** fin\_sweep\_msny\_gt gt *-- this is the filter for what gets swept*

**JOIN** enabled\_members em **ON** em.enamem\_id = gt.enamem\_id

**JOIN** current\_perm\_assignment\_vw ca **ON** ca.call\_emem\_id = gt.enamem\_id

**JOIN** fin\_msny\_funding fmf **ON** fmf.enamem\_id = gt.enamem\_id

**JOIN** (

**SELECT** g.msny\_id,

**CASE**

*-- How many days beyond planned release is the actual release?*

**WHEN** (e.enamem\_release\_date **IS** **NULL** **OR** e.enamem\_anniversary\_date **IS** **NULL**) **THEN**

**NULL** *-- return empty value to be picked up by error module*

**WHEN** (e.enamem\_release\_date - e.enamem\_anniversary\_date) > 14 **THEN**

**ADD\_MONTHS**(e.enamem\_anniversary\_date,

(**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))

+(**CASE**

**WHEN** e.enamem\_release\_date - (**ADD\_MONTHS**(e.enamem\_anniversary\_date,**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))) > 14

**THEN** 0

**ELSE** -1

**END**)

)

**WHEN** (e.enamem\_release\_date - e.enamem\_anniversary\_date) < -14 **THEN**

**ADD\_MONTHS**(e.enamem\_anniversary\_date,

(**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))

-(**CASE**

**WHEN** e.enamem\_release\_date - (**ADD\_MONTHS**(e.enamem\_anniversary\_date,**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))) < -14

**THEN** 2

**ELSE** 1

**END**)

)

**ELSE**

**ADD\_MONTHS**(e.enamem\_anniversary\_date,-1)

**END** **AS** last\_pay\_dt

**FROM** fin\_sweep\_msny\_gt g

**JOIN** enabled\_members e **ON** e.enamem\_id = g.enamem\_id

) lpm

**ON** lpm.msny\_id = gt.msny\_id

**JOIN** asg\_loc\_components alc **ON** alc.comp\_id = ca.call\_comp\_id

*-- component-level asgloc info*

**JOIN** fin\_organization\_mv cfom **ON** cfom.asgloc\_id = alc.asgloc\_id

**JOIN** assignment\_locations cal **ON** cal.asgloc\_id = alc.asgloc\_id

**JOIN** poldiv\_to\_country\_mv cpc **ON** cpc.primary\_poldivid = cfom.primarylocpoldivid

*-- "parent" asgloc info*

**JOIN** fin\_organization\_mv mfom **ON** mfom.asgloc\_id = ca.call\_aloc\_id

**JOIN** asgloc\_msn\_area\_mv mama **ON** mama.asgloc\_orgnbr = mfom.orgnbr

**JOIN** poldiv\_to\_country\_mv mpc **ON** mpc.primary\_poldivid = mfom.primarylocpoldivid

*-- "within mission" info*

**JOIN** fin\_organization\_mv wmfom **ON** wmfom.orgnbr = mama.wm\_orgnbr

*-- ecclesiastical info*

**JOIN** fin\_organization\_mv efom **ON** efom.orgnbr = mama.eccl\_orgnbr

*-- joins to get financial info for the mission*

**JOIN** fin\_poldivs mfp **ON** mfp.poldivid = mpc.country\_poldivid

**JOIN** mdm\_currency mcurr **ON** mcurr.iso\_currency\_code = mfp.finpol\_iso\_currency\_code

**JOIN** fin\_exch\_rate mpcr **ON** mpcr.from\_cur = mcurr.iso\_currency\_code

**JOIN** fin\_asglocs mfa **ON** mfa.asgloc\_id = mfom.asgloc\_id

*-- start of outer joins ---------------------------------------------*

**LEFT** **JOIN** training\_facilities tf **ON** tf.trnfac\_id = ca.call\_mtcs\_id *-- many MTC IDs are null, some 0*

**LEFT** **JOIN** col\_training\_orgs\_mv ctom **ON** ctom.orgnbr = tf.col\_orgnbr *-- must left join as well if MTC was null*

**LEFT** **JOIN** fin\_organization\_mv ffom **ON** ffom.orgnbr = fmf.fund\_unit\_id *-- some test missionaries won't have entries in fin\_organization\_mv*

**LEFT** **JOIN** fin\_organization\_mv fstake **ON** fstake.orgnbr = ffom.parentorgnbr *-- need long name of stake*

**LEFT** **JOIN** fin\_organization\_mv farea **ON** farea.orgnbr = ffom.areaorgnbr *-- need long name of area*

**LEFT** **JOIN** poldiv\_to\_country\_mv fpc **ON** fpc.primary\_poldivid = ffom.primarylocpoldivid

**LEFT** **JOIN** fin\_unit\_override fuo

**ON** fuo.orgnbr = **CASE**

**WHEN** fmf.allow\_override\_yn = 'Y' **THEN** ffom.orgnbr

**ELSE** **NULL** *-- remove ability to join to override table*

**END**

**LEFT** **JOIN** fin\_poldivs ffp *-- join for funding unit country financial attributes*

**ON** ffp.poldivid = **CASE**

**WHEN** fuo.orgnbr **IS** **NULL** **THEN** fpc.country\_poldivid

**ELSE** fuo.poldivid\_of\_equalized\_amt

**END**

*-- Now that fin\_poldivs is joined, perhaps on redirected country from poldivid\_of\_equalized\_amt,*

*-- re-join to poldiv\_to\_country\_mv to get country info.*

**LEFT** **JOIN** poldiv\_to\_country\_mv ffpc **ON** ffpc.primary\_poldivid = ffp.poldivid

**LEFT** **JOIN** mdm\_currency fcurr **ON** fcurr.iso\_currency\_code = ffp.finpol\_iso\_currency\_code

**LEFT** **JOIN** fin\_exch\_rate fpcr **ON** fpcr.from\_cur = fcurr.iso\_currency\_code

*-- Plethora of outer joins to get possible temp assignment info*

**LEFT** **JOIN** (

*-- current\_temp\_assignment\_vw didn't work since it internally used SYSDATE to filter. We*

*-- need the filter to use the sweep date, the 1st of the month.*

**SELECT** call\_id, call\_aloc\_id, call\_comp\_id, call\_emem\_id, asgmnt\_start\_date, asgmnt\_end\_date

**FROM** napi\_assignment\_mstr

**WHERE** call\_type **in** (2,3)

**AND** asgmnt\_start\_date <= i\_sweep\_dt

**AND** (asgmnt\_end\_date > i\_sweep\_dt **OR** asgmnt\_end\_date **IS** **NULL**)

**AND** status <> 7

) cta **ON** cta.call\_emem\_id = em.enamem\_id

**LEFT** **JOIN** asg\_loc\_components talc **ON** talc.comp\_id = cta.call\_comp\_id

**LEFT** **JOIN** fin\_organization\_mv tcfom **ON** tcfom.asgloc\_id = talc.asgloc\_id

**LEFT** **JOIN** assignment\_locations tcal **ON** tcal.asgloc\_id = talc.asgloc\_id

**LEFT** **JOIN** fin\_organization\_mv tmfom **ON** tmfom.asgloc\_id = cta.call\_aloc\_id

**LEFT** **JOIN** asgloc\_msn\_area\_mv tama **ON** tama.asgloc\_id = cta.call\_aloc\_id

**LEFT** **JOIN** fin\_organization\_mv twmfom **ON** twmfom.orgnbr = tama.wm\_orgnbr

**LEFT** **JOIN** fin\_organization\_mv tefom **ON** tefom.orgnbr = tama.eccl\_orgnbr

;

**END** ins\_sweep\_msny;

*/\*\*-----------------------------------------------------------------------------*

*ins\_sweep\_tally*

*Sums counts of missionary types and status by month, mission and component.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** ins\_sweep\_tally(i\_sweep\_dt **IN** **DATE**)

**IS**

l\_created\_dt **DATE** := **SYSDATE**;

**BEGIN**

**INSERT** **INTO** fin\_sweep\_tally

(

sweep\_dt, created\_dt

, comp\_id, comp\_nm

, comp\_asgloc\_orgnbr, comp\_asgloc\_orgtype\_cd, comp\_asgloc\_orgtype\_nm

, comp\_asgloc\_id, comp\_asgloc\_nm, comp\_asgloc\_sens\_flg

, asgloc\_orgnbr, asgloc\_id, asgloc\_nm

, asgloc\_area\_orgnbr, asgloc\_area\_nm

, asgloc\_country\_id, asgloc\_country\_nm, asgloc\_sens\_flg

, within\_msn\_orgnbr, within\_msn\_asgloc\_id, within\_msn\_nm, within\_msn\_sens\_flg

, eccl\_org\_orgnbr, eccl\_org\_nm, eccl\_org\_sens\_flg

, msn\_currency\_used, msn\_currency\_cd, msn\_currency\_nm, msn\_exchange\_rt

, msn\_local\_amt, msn\_us\_amt, msn\_facs\_num

, pos\_spcl\_id, prim\_pos\_nm, prim\_pos\_pros\_only\_yn

, rspbl\_org\_nm, rspbl\_orgnbr

, lang\_id, lang\_nm

, complement\_elders, complement\_sisters, complement\_couples, complement\_selders, complement\_ssisters, prefield\_elders

, prefield\_sisters, prefield\_couples, prefield\_selders, prefield\_ssisters

, mtc\_elders, mtc\_sisters, mtc\_couples, mtc\_selders, mtc\_ssisters

, infield\_elders, infield\_sisters, infield\_couples, infield\_selders, infield\_ssisters

, temp\_elders, temp\_sisters, temp\_couples, temp\_selders, temp\_ssisters

, away\_elders, away\_sisters, away\_couples, away\_selders, away\_ssisters

, lah\_couples, lah\_selders, lah\_ssisters

)

**SELECT** \* **FROM** (

**WITH** finsm **AS**

(**SELECT** \* **FROM** fin\_sweep\_msny **WHERE** sweep\_dt = i\_sweep\_dt)

**SELECT**

sweep\_dt, l\_created\_dt

, comp\_id, comp\_nm

, comp\_asgloc\_orgnbr, comp\_asgloc\_orgtype\_cd, comp\_asgloc\_orgtype\_nm

, comp\_asgloc\_id, comp\_asgloc\_nm, comp\_asgloc\_sens\_flg

, asgloc\_orgnbr, asgloc\_id, asgloc\_nm

, asgloc\_area\_orgnbr, asgloc\_area\_nm

, asgloc\_country\_id, asgloc\_country\_nm, asgloc\_sens\_flg

, within\_msn\_orgnbr, within\_msn\_asgloc\_id, within\_msn\_nm, within\_msn\_sens\_flg

, eccl\_org\_orgnbr, eccl\_org\_nm, eccl\_org\_sens\_flg

, msn\_currency\_used, msn\_currency\_cd, msn\_currency\_nm, msn\_exchange\_rt

, msn\_local\_amt, msn\_us\_amt, msn\_facs\_num

, pos\_spcl\_id, prim\_pos\_nm, prim\_pos\_pros\_only\_yn

, rspbl\_org\_nm, rspbl\_orgnbr

, lang\_id, lang\_nm

, complement\_elders, complement\_sisters, complement\_couples, complement\_selders, complement\_ssisters

, prefield\_elders, prefield\_sisters, prefield\_couples, prefield\_selders, prefield\_ssisters

, mtc\_elders, mtc\_sisters, mtc\_couples, mtc\_selders, mtc\_ssisters

, infield\_elders, infield\_sisters, infield\_couples, infield\_selders, infield\_ssisters

, **CASE** **WHEN** misstyp\_id = 1 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_elders

, **CASE** **WHEN** misstyp\_id = 2 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_sisters

, **CASE** **WHEN** misstyp\_id = 3 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_couples

, **CASE** **WHEN** misstyp\_id = 4 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_selders

, **CASE** **WHEN** misstyp\_id = 5 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_ssisters

, away\_elders, away\_sisters, away\_couples, away\_selders, away\_ssisters

, lah\_couples, lah\_selders, lah\_ssisters

**FROM** (

**SELECT**

fsm.sweep\_dt

, fsm.comp\_id, fsm.comp\_nm

, fsm.comp\_asgloc\_orgnbr, fsm.comp\_asgloc\_orgtype\_cd, fsm.comp\_asgloc\_orgtype\_nm

, fsm.comp\_asgloc\_id, fsm.comp\_asgloc\_nm, fsm.comp\_asgloc\_sens\_flg

, fsm.asgloc\_orgnbr, fsm.asgloc\_id, fsm.asgloc\_nm

, fsm.asgloc\_area\_orgnbr, fsm.asgloc\_area\_nm

, fsm.asgloc\_country\_id, fsm.asgloc\_country\_nm, fsm.asgloc\_sens\_flg

, fsm.within\_msn\_orgnbr, fsm.within\_msn\_asgloc\_id, fsm.within\_msn\_nm, fsm.within\_msn\_sens\_flg

, fsm.eccl\_org\_orgnbr, fsm.eccl\_org\_nm, fsm.eccl\_org\_sens\_flg

, fsm.msn\_currency\_used, fsm.msn\_currency\_cd, fsm.msn\_currency\_nm, fsm.msn\_exchange\_rt

, fsm.msn\_local\_amt, fsm.msn\_us\_amt, fsm.msn\_facs\_num

, acpv.posspc\_id pos\_spcl\_id, acpv.primary\_pos\_name prim\_pos\_nm, acpv.primary\_pos\_pros\_only prim\_pos\_pros\_only\_yn

, acpv.responsible\_org rspbl\_org\_nm, acpv.responsible\_orgnbr rspbl\_orgnbr

, acpv.lang\_id, acpv.language\_name lang\_nm

, alc.misstyp\_id

, **CASE** **WHEN** alc.misstyp\_id = 1 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_elders

, **CASE** **WHEN** alc.misstyp\_id = 2 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_sisters

, **CASE** **WHEN** alc.misstyp\_id = 3 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_couples

, **CASE** **WHEN** alc.misstyp\_id = 4 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_selders

, **CASE** **WHEN** alc.misstyp\_id = 5 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_ssisters

, **SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** alc.misstyp\_id = 3 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_couples

, **SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** alc.misstyp\_id = 4 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_selders

, **SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** alc.misstyp\_id = 5 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_ssisters

**FROM** finsm fsm

**JOIN** asg\_loc\_components alc **ON** alc.comp\_id = fsm.comp\_id

**JOIN** asgloc\_comp\_pos\_vw acpv **ON** acpv.comp\_id = fsm.comp\_id *-- join for position info*

**GROUP** **BY**

fsm.sweep\_dt

, fsm.comp\_id, fsm.comp\_nm

, fsm.comp\_asgloc\_orgnbr, fsm.comp\_asgloc\_orgtype\_cd, fsm.comp\_asgloc\_orgtype\_nm

, fsm.comp\_asgloc\_id, fsm.comp\_asgloc\_nm, fsm.comp\_asgloc\_sens\_flg

, fsm.asgloc\_orgnbr, fsm.asgloc\_id, fsm.asgloc\_nm

, fsm.asgloc\_area\_orgnbr, fsm.asgloc\_area\_nm

, fsm.asgloc\_country\_id, fsm.asgloc\_country\_nm, fsm.asgloc\_sens\_flg

, fsm.within\_msn\_orgnbr, fsm.within\_msn\_asgloc\_id, fsm.within\_msn\_nm, fsm.within\_msn\_sens\_flg

, fsm.eccl\_org\_orgnbr, fsm.eccl\_org\_nm, fsm.eccl\_org\_sens\_flg

, fsm.msn\_currency\_used, fsm.msn\_currency\_cd, fsm.msn\_currency\_nm, fsm.msn\_exchange\_rt

, fsm.msn\_local\_amt, fsm.msn\_us\_amt, fsm.msn\_facs\_num

, acpv.posspc\_id, acpv.primary\_pos\_name, acpv.primary\_pos\_pros\_only

, acpv.responsible\_org, acpv.responsible\_orgnbr

, acpv.lang\_id, acpv.language\_name

, alc.misstyp\_id, alc.comp\_num

) fsm

**LEFT** **OUTER** **JOIN** (**SELECT** temp\_comp\_id, **COUNT**(\*) msny\_count **FROM** finsm **GROUP** **BY** temp\_comp\_id) tfsw **ON** tfsw.temp\_comp\_id = fsm.comp\_id)

;

*-- Now grab those missionaries who are temporarily assigned to components*

*-- not used by any permanent assignment in the previous query from fin\_sweep\_msny.*

**INSERT** **INTO** fin\_sweep\_tally

(

sweep\_dt, created\_dt

, comp\_id, comp\_nm

, comp\_asgloc\_orgnbr, comp\_asgloc\_orgtype\_cd, comp\_asgloc\_orgtype\_nm

, comp\_asgloc\_id, comp\_asgloc\_nm, comp\_asgloc\_sens\_flg

, asgloc\_orgnbr, asgloc\_id, asgloc\_nm

, asgloc\_area\_orgnbr, asgloc\_area\_nm

, asgloc\_country\_id, asgloc\_country\_nm, asgloc\_sens\_flg

, within\_msn\_orgnbr, within\_msn\_asgloc\_id, within\_msn\_nm, within\_msn\_sens\_flg

, eccl\_org\_orgnbr, eccl\_org\_nm, eccl\_org\_sens\_flg

, msn\_currency\_used, msn\_currency\_cd, msn\_currency\_nm, msn\_exchange\_rt

, msn\_local\_amt, msn\_us\_amt, msn\_facs\_num

, pos\_spcl\_id, prim\_pos\_nm, prim\_pos\_pros\_only\_yn

, rspbl\_org\_nm, rspbl\_orgnbr

, lang\_id, lang\_nm

, complement\_elders, complement\_sisters, complement\_couples, complement\_selders, complement\_ssisters, prefield\_elders

, prefield\_sisters, prefield\_couples, prefield\_selders, prefield\_ssisters

, mtc\_elders, mtc\_sisters, mtc\_couples, mtc\_selders, mtc\_ssisters

, infield\_elders, infield\_sisters, infield\_couples, infield\_selders, infield\_ssisters

, temp\_elders, temp\_sisters, temp\_couples, temp\_selders, temp\_ssisters

, away\_elders, away\_sisters, away\_couples, away\_selders, away\_ssisters

, lah\_couples, lah\_selders, lah\_ssisters

)

**SELECT** \* **FROM**

(

**WITH** ot **AS** *-- "ot" stands for orphaned temporaries*

(

**SELECT** fsm.temp\_comp\_id, fsm.temp\_comp\_nm,

fsm.temp\_comp\_asgloc\_orgnbr, fsm.temp\_comp\_asgloc\_id, fsm.temp\_comp\_asgloc\_nm, fsm.temp\_comp\_asgloc\_sens\_flg,

fsm.temp\_asgloc\_orgnbr, fsm.temp\_asgloc\_id, fsm.temp\_asgloc\_nm, fsm.temp\_asgloc\_sens\_flg,

fsm.temp\_within\_msn\_orgnbr, fsm.temp\_within\_msn\_asgloc\_id, fsm.temp\_within\_msn\_nm, fsm.temp\_within\_msn\_sens\_flg,

fsm.temp\_eccl\_org\_orgnbr, fsm.temp\_eccl\_org\_nm, fsm.temp\_eccl\_org\_sens\_flg,

**SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** fsm.misstyp\_id = 3 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_couples,

**SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** fsm.misstyp\_id = 4 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_selders,

**SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** fsm.misstyp\_id = 5 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_ssisters,

**COUNT**(\*) **AS** cnt

**FROM** fin\_sweep\_msny fsm

**WHERE** fsm.sweep\_dt = i\_sweep\_dt

**AND** fsm.temp\_comp\_id **IS** **NOT** **NULL**

**AND** **NOT** **EXISTS** (**SELECT** **NULL**

**FROM** fin\_sweep\_tally fst

**WHERE** fst.sweep\_dt = i\_sweep\_dt

**AND** fst.comp\_id = fsm.temp\_comp\_id)

**GROUP** **BY**

fsm.temp\_comp\_id, fsm.temp\_comp\_nm,

fsm.temp\_comp\_asgloc\_orgnbr, fsm.temp\_comp\_asgloc\_id, fsm.temp\_comp\_asgloc\_nm, fsm.temp\_comp\_asgloc\_sens\_flg,

fsm.temp\_asgloc\_orgnbr, fsm.temp\_asgloc\_id, fsm.temp\_asgloc\_nm, fsm.temp\_asgloc\_sens\_flg,

fsm.temp\_within\_msn\_orgnbr, fsm.temp\_within\_msn\_asgloc\_id, fsm.temp\_within\_msn\_nm, fsm.temp\_within\_msn\_sens\_flg,

fsm.temp\_eccl\_org\_orgnbr, fsm.temp\_eccl\_org\_nm, fsm.temp\_eccl\_org\_sens\_flg

)

**SELECT**

i\_sweep\_dt, l\_created\_dt

, ot.temp\_comp\_id **AS** comp\_id, ot.temp\_comp\_nm **AS** comp\_nm

, ot.temp\_comp\_asgloc\_orgnbr **AS** comp\_asgloc\_orgnbr, cfom.orgtypecode **AS** comp\_asgloc\_orgtype\_cd, cfom.orgtypeshortdesctxt **AS** comp\_asgloc\_orgtype\_nm

, ot.temp\_comp\_asgloc\_id **AS** comp\_asgloc\_id, ot.temp\_comp\_asgloc\_nm **AS** comp\_asgloc\_nm, ot.temp\_comp\_asgloc\_sens\_flg **AS** comp\_asgloc\_sens\_flg

, ot.temp\_asgloc\_orgnbr **AS** asgloc\_orgnbr, ot.temp\_asgloc\_id **AS** asgloc\_id, ot.temp\_asgloc\_nm **AS** asgloc\_nm

, acom.areaorgnbr **AS** asgloc\_area\_orgnbr, acom.areamediumname **AS** asgloc\_area\_nm

, apc.country\_poldivid asgloc\_country\_id, apc.country\_longname **AS** asgloc\_country\_nm, ot.temp\_asgloc\_sens\_flg **AS** asgloc\_sens\_flg

, ot.temp\_within\_msn\_orgnbr **AS** within\_msn\_orgnbr, ot.temp\_within\_msn\_asgloc\_id **AS** within\_msn\_asgloc\_id, ot.temp\_within\_msn\_nm **AS** within\_msn\_nm, ot.temp\_within\_msn\_sens\_flg **AS** within\_msn\_sens\_flg

, ot.temp\_eccl\_org\_orgnbr **AS** eccl\_org\_orgnbr, ot.temp\_eccl\_org\_nm **AS** eccl\_org\_nm, ot.temp\_eccl\_org\_sens\_flg **AS** eccl\_org\_sens\_flg

, afp.cfis\_local\_or\_us\_flag **AS** msn\_currency\_used, afp.finpol\_iso\_currency\_code **AS** msn\_currency\_cd

, acurr.currency\_desc **AS** msn\_currency\_nm, apcr.exch\_rate **AS** msn\_exchange\_rt

, **CASE** **WHEN** afp.cfis\_local\_or\_us\_flag = 'L' **THEN** afa.base\_amt **ELSE** **ROUND**(afa.base\_amt / apcr.exch\_rate) **END** **AS** msn\_local\_amt

, **CASE** **WHEN** afp.cfis\_local\_or\_us\_flag = 'L' **THEN** **ROUND**(afa.base\_amt \* apcr.exch\_rate) **ELSE** afa.base\_amt **END** **AS** msn\_us\_amt

, afa.gl\_unit\_num **AS** msn\_facs\_num

, acpv.posspc\_id pos\_spcl\_id, acpv.primary\_pos\_name prim\_pos\_nm, acpv.primary\_pos\_pros\_only prim\_pos\_pros\_only\_yn

, acpv.responsible\_org rspbl\_org\_nm, acpv.responsible\_orgnbr rspbl\_orgnbr

, acpv.lang\_id, acpv.language\_name lang\_nm

, **CASE** **WHEN** alc.misstyp\_id = 1 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_elders

, **CASE** **WHEN** alc.misstyp\_id = 2 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_sisters

, **CASE** **WHEN** alc.misstyp\_id = 3 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_couples

, **CASE** **WHEN** alc.misstyp\_id = 4 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_selders

, **CASE** **WHEN** alc.misstyp\_id = 5 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_ssisters

*-- Missionaries with active temp assignments will always be infield. They can have planned*

*-- temp assignments while prefield or MTC, but those assignments won't show up*

*-- in current temp assignment inline view, which is used in the initial insert to fin\_sweep\_msny*

, 0 **AS** prefield\_elders

, 0 **AS** prefield\_sisters

, 0 **AS** prefield\_couples

, 0 **AS** prefield\_selders

, 0 **AS** prefield\_ssisters

, 0 **AS** mtc\_elders

, 0 **AS** mtc\_sisters

, 0 **AS** mtc\_couples

, 0 **AS** mtc\_selders

, 0 **AS** mtc\_ssisters

, 0 **AS** infield\_elders

, 0 **AS** infield\_sisters

, 0 **AS** infield\_couples

, 0 **AS** infield\_selders

, 0 **AS** infield\_ssisters

, **CASE** **WHEN** alc.misstyp\_id = 1 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_elders

, **CASE** **WHEN** alc.misstyp\_id = 2 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_sisters

, **CASE** **WHEN** alc.misstyp\_id = 3 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_couples

, **CASE** **WHEN** alc.misstyp\_id = 4 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_selders

, **CASE** **WHEN** alc.misstyp\_id = 5 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_ssisters

, 0 **AS** away\_elders

, 0 **AS** away\_sisters

, 0 **AS** away\_couples

, 0 **AS** away\_selders

, 0 **AS** away\_ssisters

, ot.lah\_couples

, ot.lah\_selders

, ot.lah\_ssisters

**FROM** ot

**JOIN** assignment\_locations al

**ON** al.asgloc\_id = ot.temp\_asgloc\_id

**JOIN** asg\_loc\_components alc

**ON** alc.comp\_id = ot.temp\_comp\_id

**JOIN** assignment\_locations cal

**ON** cal.asgloc\_id = alc.asgloc\_id

**JOIN** fin\_organization\_mv cfom *-- join for component-level assignment location info*

**ON** cfom.orgnbr = cal.asgloc\_col\_orgnbr

**JOIN** asgloc\_comp\_pos\_vw acpv *-- join for position info*

**ON** acpv.comp\_id = ot.temp\_comp\_id

**JOIN** fin\_organization\_mv acom *-- join for assignment location info*

**ON** acom.orgnbr = al.asgloc\_col\_orgnbr

**JOIN** poldiv\_to\_country\_mv apc *-- join for assignment location country*

**ON** apc.primary\_poldivid = acom.primarylocpoldivid

**JOIN** fin\_poldivs afp *-- join for assignment location country financial attributes*

**ON** afp.poldivid = apc.country\_poldivid

**JOIN** currency\_type\_vw acurr

**ON** acurr.iso\_currency\_code = afp.finpol\_iso\_currency\_code

**JOIN** fin\_exch\_rate apcr

**ON** apcr.from\_cur = acurr.iso\_currency\_code

**JOIN** fin\_asglocs afa

**ON** afa.asgloc\_id = al.asgloc\_id

);

**END** ins\_sweep\_tally;

*/\*\*-----------------------------------------------------------------------------*

*ins\_pay\_hist*

*Captures records for each missionary funded by a funding unit.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** ins\_pay\_hist(i\_sweep\_dt **IN** **DATE**)

**IS**

**BEGIN**

**INSERT** **INTO** fin\_pay\_hist

(msny\_id

,legacy\_miss\_id

,fund\_unit\_id

,fund\_type\_id

,pay\_dt

,amount

,pay\_nbr

,dr\_cr\_code

,cfar\_comments

,mod\_by

,mod\_dt

,adj\_yn

,currency\_cd)

**SELECT** msny\_id

,legacy\_miss\_id

,fund\_unit\_id

,fund\_type\_id

,sweep\_dt

,fund\_unit\_local\_amt

,crnt\_pmt\_num

,'DR'

,'Monthly Sweep'

,'SYSTEM'

,**SYSDATE**

,**NULL** **AS** adj\_yn *-- system-created pay hist rows use NULL. Y reserved for human adjustments.*

,fund\_unit\_currency\_cd **AS** currency\_cd

**FROM** fin\_sweep\_msny

**WHERE** sweep\_dt = i\_sweep\_dt

**AND** blbl\_month\_yn = 'Y'

**AND** (misstyp\_id **IN** (1,2) **OR** (misstyp\_id = 3 **AND** fund\_unit\_local\_amt > 0 ))

**ORDER** **BY** fund\_unit\_id, call\_type\_id, procstat\_id **DESC**, msny\_id;

**END** ins\_pay\_hist;

*/\*\*-----------------------------------------------------------------------------*

*run\_error\_check*

*Creates error records in FIN\_SWEEP\_MSNY\_ERR for each anomalous condition it*

*finds.*

*The error table is cleared out each time the error check is run. Then the*

*various checks are run, writing one record for each missionary sporting the*

*error condition.*

*Will be run each time the monthly sweep is run.*

*Can be run independently if needed.*

*Can even be run after a sweep has been finalized (assuming a fresh view of this*

*month's acceptable errors is required).*

*%param i\_sweep\_dt The effective date of the run. The provided date will*

*be truncated to the first of the month.*

*------------------------------------------------------------------------------\*/*

**PROCEDURE** run\_error\_check(i\_sweep\_dt **IN** **DATE** **DEFAULT** **SYSDATE**)

**IS**

l\_sweep\_dt **DATE** := **TRUNC**(**NVL**(i\_sweep\_dt,**SYSDATE**),'Month');

l\_marker **INTEGER** := 0;

l\_count **INTEGER** := 0;

**BEGIN**

logs.dbg('Clean out the error table');

**DELETE** **FROM** fin\_sweep\_msny\_err;

*-- If being called independently, populate the sweep set first*

**SELECT** **COUNT**(\*) **INTO** l\_count **FROM** fin\_sweep\_msny\_gt;

**IF** (l\_count = 0) **THEN**

load\_sweep\_set(l\_sweep\_dt);

**END** **IF**;

*-- missing mission start date*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Missing mission start (MTC) date.'

,'ENABLED\_MEMBER\_MSTR'

,'ENAMEM\_MISSION\_START\_DATE'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**WHERE** em.enamem\_mission\_start\_date **IS** **NULL**;

*-- missing release date*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Missing release date.'

,'ENABLED\_MEMBER\_MSTR'

,'ENAMEM\_RELEASE\_DATE'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**WHERE** em.enamem\_release\_date **IS** **NULL**;

*-- missing anniversary date*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Missing anniversary date.'

,'ENABLED\_MEMBER\_MSTR'

,'ENAMEM\_ANNIVERSARY\_DATE'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**WHERE** em.enamem\_anniversary\_date **IS** **NULL**;

*-- anniversary date before mission start date*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Anniversary before start date.'

,'ENABLED\_MEMBER\_MSTR'

,'ENAMEM\_ANNIVERSARY\_DATE'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**WHERE** em.enamem\_anniversary\_date < em.enamem\_mission\_start\_date;

*-- sweep missionaries that don't have assignments*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm)

**SELECT** gt.msny\_id

,'Missing assignment.'

,'NAPI\_ASSIGNMENT\_MSTR'

**FROM** fin\_sweep\_msny\_gt gt

**LEFT** **JOIN** current\_perm\_assignment\_vw ca

**ON** ca.call\_emem\_id = gt.enamem\_id

**WHERE** ca.call\_emem\_id **IS** **NULL**;

*-- missing home unit*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** m.ID **AS** msny\_id

,'Missing home unit.'

,'ENABLED\_MEMBER\_MSTR'

,'ENAMEM\_HOME\_UNIT\_ID'

**FROM** missionaries m

**JOIN** enabled\_members em

**ON** em.enamem\_id = m.enamem\_id

**WHERE** em.procstat\_id **IN** (11, 12, 13, 14)

**AND** em.enamem\_release\_date >= l\_sweep\_dt

**AND** em.enamem\_home\_unit\_id **IS** **NULL**;

*-- has assignment, but start date is missing*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Missing assignment start date.'

,'NAPI\_ASSIGNMENT\_MSTR'

,'ASGMNT\_START\_DATE'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**JOIN** current\_perm\_assignment\_vw ca

**ON** ca.call\_emem\_id = em.enamem\_id

**WHERE** ca.asgmnt\_start\_date **IS** **NULL**;

*-- has assignment, but end date is missing*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Missing assignment end date.'

,'NAPI\_ASSIGNMENT\_MSTR'

,'ASGMNT\_END\_DATE'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**JOIN** current\_perm\_assignment\_vw ca

**ON** ca.call\_emem\_id = em.enamem\_id

**WHERE** ca.asgmnt\_end\_date **IS** **NULL**;

*-- missionaries that don't have funding record*

*-- all missionaries going on, or on a mission should have a funding record*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm)

**SELECT** gt.msny\_id

,'Missing funding record.'

,'FIN\_MSNY\_FUNDING'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**LEFT** **JOIN** fin\_msny\_funding fmf

**ON** fmf.enamem\_id = em.enamem\_id

**WHERE** fmf.enamem\_id **IS** **NULL**;

*-- has funding record, but don't have a funding unit*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Missing funding unit.'

,'FIN\_MSNY\_FUNDING'

,'FUND\_UNIT\_ID'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** enabled\_members em

**ON** em.enamem\_id = gt.enamem\_id

**JOIN** fin\_msny\_funding fmf

**ON** fmf.enamem\_id = em.enamem\_id

**WHERE** fmf.fund\_type\_id **IN** (1, 2, 4, 5)

**AND** fmf.fund\_unit\_id **IS** **NULL**;

*-- Missionary is assigned to inactive (Deactivated) components*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** fsm.msny\_id

,'Missionary assigned to deactivated component ' || fsm.comp\_nm ||

' in ' || fsm.comp\_asgloc\_nm || ' (comp\_id=' || fsm.comp\_id || ').'

,'NAPI\_ASSIGNMENT\_MSTR'

,'COMP\_ID'

**FROM** fin\_sweep\_msny fsm

**JOIN** asg\_loc\_components alc

**ON** alc.comp\_id = fsm.comp\_id

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** alc.compsta\_id = 3;

*-- Missionary is assigned to inactive (Discontinued) assignment location*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** fsm.msny\_id

,'Missionary assigned to discontinued assignment location ' ||

fsm.comp\_asgloc\_nm || ' (asgloc\_id=' || fsm.comp\_asgloc\_id || ').'

,'NAPI\_ASSIGNMENT\_MSTR'

,'CALL\_ALOC\_ID'

**FROM** fin\_sweep\_msny fsm

**JOIN** assignment\_locations al

**ON** al.asgloc\_id = fsm.comp\_asgloc\_id

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** al.asglocsta\_id = 5;

*-- Assignment location not found in CDOL anymore*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Assignment location not found in CDOL (orgnbr='||al.asgloc\_col\_orgnbr||', name='||al.name||')'

,'ASSIGNMENT\_LOCATIONS'

,'ASGLOC\_COL\_ORGNBR'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** current\_perm\_assignment\_vw ca

**ON** ca.call\_emem\_id = gt.enamem\_id

**JOIN** assignment\_locations al

**ON** al.asgloc\_id = ca.call\_aloc\_id

**LEFT** **JOIN** fin\_organization\_mv mcom

**ON** mcom.orgnbr = al.asgloc\_col\_orgnbr

**WHERE** mcom.orgid **IS** **NULL**;

*-- Mission Country not in our Poldiv to Country translator*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm)

**SELECT** gt.msny\_id

,'Mission country ('||mcom.primarylocpoldivid||') missing from POLDIV\_TO\_COUNTRY\_MV.'

,'POLDIV\_TO\_COUNTRY\_MV'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** current\_perm\_assignment\_vw ca

**ON** ca.call\_emem\_id = gt.enamem\_id

**JOIN** assignment\_locations al

**ON** al.asgloc\_id = ca.call\_aloc\_id

**JOIN** fin\_organization\_mv mcom

**ON** mcom.orgnbr = al.asgloc\_col\_orgnbr

**LEFT** **JOIN** poldiv\_to\_country\_mv mpc

**ON** mpc.primary\_poldivid = mcom.primarylocpoldivid

**WHERE** mpc.primary\_poldivid **IS** **NULL**;

*-- Mission area not found in CDOL*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** fsm.msny\_id

,'Mission area is blank in CDOL for mission (orgnbr='||fsm.asgloc\_orgnbr||').'

,'COL\_ORGANIZATION\_MV'

,'AREAORGNBR'

**FROM** fin\_sweep\_msny fsm

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** fsm.asgloc\_orgnbr **IS** **NOT** **NULL**

**AND** fsm.asgloc\_area\_orgnbr **IS** **NULL**;

*-- Funding unit Country not in our Poldiv to Country translator*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** gt.msny\_id

,'Cannot get country from funding unit ('||fmf.fund\_unit\_id||').'

,'COL\_ORGANIZATION\_MV'

,'PRIMARYLOCPOLDIVID'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** fin\_msny\_funding fmf

**ON** fmf.enamem\_id = gt.enamem\_id

**JOIN** fin\_organization\_mv fcom

**ON** fcom.orgnbr = fmf.fund\_unit\_id

**LEFT** **JOIN** poldiv\_to\_country\_mv fpc

**ON** fpc.primary\_poldivid = fcom.primarylocpoldivid

**WHERE** fmf.fund\_type\_id **IN** (1, 2, 4, 5)

**AND** fpc.primary\_poldivid **IS** **NULL**;

*-- Funding unit Country not in our Poldiv to Country translator*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm)

**SELECT** gt.msny\_id

,'Funding unit country (primary\_poldivid='||fcom.primarylocpoldivid||') missing from POLDIV\_TO\_COUNTRY\_MV.'

,'POLDIV\_TO\_COUNTRY\_MV'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** fin\_msny\_funding fmf

**ON** fmf.enamem\_id = gt.enamem\_id

**JOIN** fin\_organization\_mv fcom

**ON** fcom.orgnbr = fmf.fund\_unit\_id

**LEFT** **JOIN** poldiv\_to\_country\_mv fpc

**ON** fpc.primary\_poldivid = fcom.primarylocpoldivid

**WHERE** fmf.fund\_type\_id **IN** (1, 2, 4, 5)

**AND** fcom.primarylocpoldivid **IS** **NOT** **NULL**

**AND** fpc.primary\_poldivid **IS** **NULL**;

*-- Mission country not in our poldiv financial attributes tables*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm)

**SELECT** gt.msny\_id

,'Mission country '||mpc.country\_poldivid||' missing from FIN\_POLDIVS.'

,'FIN\_POLDIVS'

**FROM** fin\_sweep\_msny\_gt gt

**JOIN** current\_perm\_assignment\_vw ca

**ON** ca.call\_emem\_id = gt.enamem\_id

**JOIN** assignment\_locations al

**ON** al.asgloc\_id = ca.call\_aloc\_id

**JOIN** fin\_organization\_mv mcom

**ON** mcom.orgnbr = al.asgloc\_col\_orgnbr

**JOIN** poldiv\_to\_country\_mv mpc *-- join for mission country*

**ON** mpc.primary\_poldivid = mcom.primarylocpoldivid

**LEFT** **JOIN** fin\_poldivs mfp *-- join for mission currency*

**ON** mfp.poldivid = mpc.country\_poldivid

**WHERE** mfp.poldivid **IS** **NULL**;

*-- Downstream processing needs stake of funding unit*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt

,table\_nm

,column\_nm)

**SELECT** fsm.msny\_id

,'Cannot get stake from funding unit ('||fsm.fund\_unit\_id||'). Funding unit is '||com.orgstatuscodedesc||'.'

,'COL\_ORGANIZATION\_MV'

,'PARENTORGNBR'

**FROM** fin\_sweep\_msny fsm

**JOIN** fin\_organization\_mv com **ON** com.orgnbr = fsm.fund\_unit\_id

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** fsm.fund\_type\_id **IN** (1, 2, 4, 5)

**AND** fsm.fund\_unit\_stake\_id **IS** **NULL**;

*-- Downstream processing needs country of funding unit*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt)

**SELECT** fsm.msny\_id

,'Cannot get country from funding unit ('||fsm.fund\_unit\_id||')'||**CHR**(10)||

'col\_organization\_mv.primarylocpoldivid='||fcom.primarylocpoldivid||**CHR**(10)||

'poldiv\_to\_country\_mv.country\_poldivid='||ptc.country\_poldivid||**CHR**(10)||

'fin\_poldivs.poldivid='||ffp.poldivid

**FROM** fin\_sweep\_msny fsm

**LEFT** **JOIN** fin\_organization\_mv fcom

**ON** fcom.orgnbr = fsm.fund\_unit\_id

**LEFT** **JOIN** poldiv\_to\_country\_mv ptc

**ON** ptc.primary\_poldivid = fcom.primarylocpoldivid

**LEFT** **JOIN** fin\_poldivs ffp

**ON** ffp.poldivid = ptc.country\_poldivid

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** fsm.fund\_type\_id **IN** (1, 2, 4, 5)

**AND** fsm.fund\_unit\_country\_id **IS** **NULL**;

*-- Currency from funding unit*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt)

**SELECT** fsm.msny\_id

,'Cannot get currency from funding unit ('||fsm.fund\_unit\_id||')'||**CHR**(10)||

'col\_organization\_mv.primarylocpoldivid='||fcom.primarylocpoldivid||**CHR**(10)||

'poldiv\_to\_country\_mv.country\_poldivid='||ptc.country\_poldivid||**CHR**(10)||

'fin\_poldivs.finpol\_iso\_currency\_code='||ffp.finpol\_iso\_currency\_code

**FROM** fin\_sweep\_msny fsm

**LEFT** **JOIN** fin\_organization\_mv fcom

**ON** fcom.orgnbr = fsm.fund\_unit\_id

**LEFT** **JOIN** poldiv\_to\_country\_mv ptc

**ON** ptc.primary\_poldivid = fcom.primarylocpoldivid

**LEFT** **JOIN** fin\_poldivs ffp

**ON** ffp.poldivid = ptc.country\_poldivid

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** fsm.fund\_type\_id **IN** (1, 2, 4, 5)

**AND** fsm.fund\_unit\_currency\_cd **IS** **NULL**;

*-- Currency from mission (should be non-zero rarely or never)*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt)

**SELECT** fsm.msny\_id

,'Cannot get currency from mission (orgnbr='||fsm.asgloc\_orgnbr||')'

**FROM** fin\_sweep\_msny fsm

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** fsm.asgloc\_orgnbr **IS** **NOT** **NULL**

**AND** fsm.msn\_currency\_cd **IS** **NULL**;

*-- Country from mission (should be non-zero rarely or never)*

l\_marker := l\_marker + 1;

logs.dbg('Sweep Error Check #'||l\_marker);

**INSERT** **INTO** fin\_sweep\_msny\_err

(msny\_id

,err\_txt)

**SELECT** fsm.msny\_id

,'Cannot get country from mission (orgnbr='||fsm.asgloc\_orgnbr||')'

**FROM** fin\_sweep\_msny fsm

**WHERE** fsm.sweep\_dt = l\_sweep\_dt

**AND** fsm.asgloc\_orgnbr **IS** **NOT** **NULL**

**AND** fsm.asgloc\_country\_id **IS** **NULL**;

**END** run\_error\_check;

*--------------------------------------------------------------------------------*

*-- PUBLIC FUNCTIONS AND PROCEDURES*

*--------------------------------------------------------------------------------*

*--------------------------------------------------------------------------------*

**PROCEDURE** snap\_exchange\_rate\_map

(

i\_snap\_dt **IN** **DATE** **DEFAULT** **SYSDATE**

)

**IS**

**CURSOR** cur\_diffs(i\_date **IN** **DATE**) **IS**

**SELECT** fer.fin\_exch\_rate\_id

, fer.from\_cur, fer.exch\_rate, fer.eff\_dt

, psr.exch\_rate **AS** new\_exch\_rate, psr.eff\_dt **AS** new\_eff\_dt

**FROM** fin\_exch\_rate fer

*-- inline view to get latest rates from PeopleSoft copy*

**JOIN** (**SELECT** from\_cur, to\_cur, cur\_exchng\_rt **AS** exch\_rate, effdt **AS** eff\_dt

**FROM** (**SELECT** from\_cur, to\_cur, cur\_exchng\_rt, effdt

, **ROW\_NUMBER**() **OVER** (**PARTITION** **BY** from\_cur, to\_cur **ORDER** **BY** effdt **DESC**) rn *-- determine latest rate*

**FROM** ps\_cur\_rt\_tbl\_mv

**WHERE** effdt <= i\_date *-- filter out PS rates newer than the 1st of the month*

**AND** cur\_rt\_type = 'CRRNT' *-- only Current exchange rates*

**AND** to\_cur = 'USD')

**WHERE** rn = 1

) psr **ON** psr.from\_cur = fer.from\_cur

**WHERE** fer.exch\_rate <> psr.exch\_rate

**AND** psr.eff\_dt > fer.eff\_dt;

*-- although meant to only be called by the monthly sweep job, if called manually*

*-- at other times in the month, we force it to the month start anyway*

l\_snap\_dt **DATE** := **TRUNC**(**NVL**(i\_snap\_dt,**SYSDATE**),'Month');

l\_rows **INTEGER** := 0;

**BEGIN**

logs.dbg('Updating copy of latest PeopleSoft/CIFS exchange rates for USD conversions...');

**FOR** lr **IN** cur\_diffs(l\_snap\_dt) **LOOP**

**UPDATE** fin\_exch\_rate

**SET** exch\_rate = lr.new\_exch\_rate

, eff\_dt = lr.new\_eff\_dt

**WHERE** fin\_exch\_rate\_id = lr.fin\_exch\_rate\_id;

l\_rows := l\_rows + 1;

**END** **LOOP**;

logs.dbg('Updated '||l\_rows||' PS exchange rate mappings...');

**COMMIT**;

**END** snap\_exchange\_rate\_map;

*--------------------------------------------------------------------------------*

**PROCEDURE** run\_sweep

(

i\_sweep\_dt **IN** **DATE** **DEFAULT** **SYSDATE**

) **IS**

*-- although they can run the sweep at any time, we force it to the month start*

l\_sweep\_dt **DATE** := **TRUNC**(**NVL**(i\_sweep\_dt,**SYSDATE**),'Month');

l\_proc\_hist\_id fin\_proc\_hist.proc\_hist\_id%**TYPE** := 0;

**BEGIN**

*-- Front end should be setting the client id to the user's login upon connection to the DB pool.*

*-- When the front end logs into the database, it is through a PTC account which*

*-- has an AFTER LOGON trigger that sets the app code.*

*-- The monthly job that calls this routine also sets the client ID and app code.*

*-- If you are attempting to call this routine in anonymous PL/SQL block, either do so*

*-- by logging into the PTC account first, or set the client ID and app code*

*-- manually. See the "what" text for the Monthly Sweep job as an example.*

*-- Validate parameter assumptions*

excp.assert(l\_sweep\_dt = **TRUNC**(**SYSDATE**,'Month'),'Unsupported use. It is currently not allowed to run a sweep for a prior or future month.');

*-- Ensure sweep for this month isn't already finalized*

excp.assert(sweep\_finalized(l\_sweep\_dt) = **FALSE**,'Sweep for this month has already been finalized. Cannot run again.');

logs.info('BEGIN');

*-- Add row to process history table*

tag\_proc\_hist(l\_sweep\_dt, gc\_start, l\_proc\_hist\_id, 'RUNNING', 'Sweep Started.');

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Cleaning up prior run for '||**TO\_CHAR**(l\_sweep\_dt,'YYYYMonDD'));

clean\_prior\_sweep(l\_sweep\_dt);

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Load temp table with sweep set');

load\_sweep\_set(l\_sweep\_dt);

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Adjusting fund\_type\_id');

*-- Keep fund\_type\_id on fin\_msny\_funding updated with changes to countries and units (fin\_poldivs and fin\_church\_units)*

adj\_msny\_fund\_type\_id();

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Adding detail data to Sweep Missionary table');

ins\_sweep\_msny(l\_sweep\_dt);

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Running error diagnostics');

run\_error\_check(l\_sweep\_dt);

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Inserting into Sweep Tally table');

ins\_sweep\_tally(l\_sweep\_dt);

**COMMIT**; *-- we decided to let the driver be in charge of the commit, instead of the caller.*

*-- Cap off row in process history table*

tag\_proc\_hist(l\_sweep\_dt, gc\_end, l\_proc\_hist\_id, 'SUCCESS', 'Sweep Ended.');

logs.info('END');

**EXCEPTION**

**WHEN** **OTHERS** **THEN**

tag\_proc\_hist(l\_sweep\_dt, gc\_end, l\_proc\_hist\_id, 'ERROR', **SQLERRM**);

**RAISE**;

**END** run\_sweep;

*--------------------------------------------------------------------------------*

**PROCEDURE** finalize\_sweep

(

i\_proc\_hist\_id **IN** fin\_proc\_hist.proc\_hist\_id%**TYPE**

) **IS**

l\_sweep\_dt **DATE**;

l\_proc\_hist\_id fin\_proc\_hist.proc\_hist\_id%**TYPE**;

**FUNCTION** get\_sweep\_dt **RETURN** **DATE**

**IS**

l\_sweep\_dt **DATE**;

**BEGIN**

**SELECT** proc\_dt

**INTO** l\_sweep\_dt

**FROM** fin\_proc\_hist

**WHERE** proc\_hist\_id = i\_proc\_hist\_id;

**RETURN** l\_sweep\_dt;

**EXCEPTION**

**WHEN** **NO\_DATA\_FOUND** **THEN**

logs.msg(msgs.ERROR\_MSG\_CD,cnst.ERROR,'proc\_hist\_id: '||i\_proc\_hist\_id||

' is not a valid value.',**TRUE**);

**END** get\_sweep\_dt;

**BEGIN**

*-- If job did not set, then make sure a client ID is set. Front end should be*

*-- setting the client id to the user's login upon connection to the DB pool.*

**IF** (env.get\_client\_id **IS** **NULL**) **THEN**

env.init\_client\_ctx(i\_client\_id => **USER**, i\_app\_cd => 'RCRD');

**END** **IF**;

*-- Validate parameter assumptions*

excp.assert(i\_proc\_hist\_id **IS** **NOT** **NULL**,

'The ID of latest successful sweep record in FIN\_PROC\_HIST must be supplied.');

l\_sweep\_dt := get\_sweep\_dt;

*-- Validate parameter assumptions*

excp.assert(l\_sweep\_dt = **TRUNC**(**SYSDATE**,'Month'),

'Unsupported use. Cannot finalize a sweep for a prior or future month.');

l\_proc\_hist\_id := i\_proc\_hist\_id;

*-- Ensure sweep for this month isn't already finalized*

excp.assert(sweep\_finalized(l\_sweep\_dt) = **FALSE**,'Sweep for '||

**TO\_CHAR**(l\_sweep\_dt,'Month YYYY')||' has already been finalized. Cannot finalize again.');

logs.info('BEGIN');

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Removing prior payments for this month');

clean\_prior\_payments(l\_sweep\_dt);

tag\_proc\_hist(l\_sweep\_dt, gc\_update, l\_proc\_hist\_id, **NULL**, 'Inserting to Payment History table');

ins\_pay\_hist(l\_sweep\_dt);

**COMMIT**; *-- we decided to let the driver be in charge of the commit, instead of the caller.*

*-- Cap off row in process history table*

tag\_proc\_hist(l\_sweep\_dt, gc\_end, l\_proc\_hist\_id, 'FINALIZED', 'Sweep Finalized.');

logs.info('END');

env.reset\_client\_ctx;

**EXCEPTION**

**WHEN** **OTHERS** **THEN**

tag\_proc\_hist(l\_sweep\_dt, gc\_end, l\_proc\_hist\_id, 'ERROR', **SQLERRM**);

**RAISE**;

**END** finalize\_sweep;

*--------------------------------------------------------------------------------*

**PROCEDURE** weekly\_tally(p\_run\_date **DATE** **DEFAULT** **SYSDATE** ) **AS**

lv\_run\_date **DATE**;

**BEGIN**

lv\_run\_date := **TRUNC**(p\_run\_date);

**EXECUTE** **IMMEDIATE** 'TRUNCATE TABLE msny\_week\_tally\_cand\_gt';

*/\*\*\* list of missionaries to process \*\*\*/*

**INSERT** **INTO** msny\_week\_tally\_cand\_gt

**SELECT** m.id

,m.enamem\_id

,m.legacy\_miss\_id

,m.mrn

,mn.first\_nm||**DECODE**(em.misstyp\_id,3,' and '||mnw.first\_nm,**NULL**)

,mn.last\_nm

,**DECODE**(em.misstyp\_id,3,**NULL**,mn.middle\_nm)

,em.term\_months pmts\_to\_send

,**SUM**(**DECODE**(fph.dr\_cr\_code, 'DR', 1, 'CR', -1)) pmts\_sent

**FROM** enabled\_members em

**JOIN** missionaries m **ON** m.legacy\_miss\_id = em.legacy\_miss\_id *-- only joins husband for couples*

**JOIN** current\_perm\_assignment\_vw cpa **ON** cpa.call\_emem\_id = em.enamem\_id **AND** cpa.early\_release\_flag <> 'Y' *-- RMH Records-4102 Filter out early released missionaries*

**JOIN** msny\_name mn **ON** mn.legacy\_miss\_id = em.legacy\_miss\_id **AND** mn.name\_type\_id = 10 *-- Candidate name*

**LEFT** **JOIN** msny\_name mnw **ON** mnw.legacy\_miss\_id = em.legacy\_spouse\_miss\_id **AND** mnw.name\_type\_id = 10 *-- Wife's Candidate name*

**LEFT** **JOIN** fin\_pay\_hist fph **ON** fph.legacy\_miss\_id = m.legacy\_miss\_id

**WHERE** em.procstat\_id **IN** (11, 12, 13, 14)

**AND** em.enamem\_release\_date >= lv\_run\_date

**GROUP** **BY** m.id

,m.enamem\_id

,m.legacy\_miss\_id

,m.mrn

,mn.first\_nm||**DECODE**(em.misstyp\_id,3,' and '||mnw.first\_nm,**NULL**)

,mn.last\_nm

,**DECODE**(em.misstyp\_id,3,**NULL**,mn.middle\_nm)

,em.term\_months;

**COMMIT**;

*/\*\*\*\* bring together temp msny, fund, and serving info \*\*\*\*/*

**EXECUTE** **IMMEDIATE** 'TRUNCATE TABLE msny\_week\_tally\_gt';

**INSERT** **INTO** msny\_week\_tally\_gt

( sweep\_msny\_id

, sweep\_dt

,created\_dt

,msny\_id

,legacy\_miss\_id

,enamem\_id

,citizenship\_id

,procstat\_id

,msny\_nm

,misstyp\_id

,term\_months

,live\_at\_home\_yn

,mtc\_dt

,arriv\_dt

,release\_dt

,anniv\_dt

,mtc\_id

,mtc\_cd

,mtc\_nm

,call\_type\_id

,comp\_id

,comp\_nm

,comp\_asgloc\_orgnbr

,comp\_asgloc\_id

,comp\_asgloc\_nm

,comp\_asgloc\_orgtype\_cd

,comp\_asgloc\_orgtype\_nm

,comp\_asgloc\_country\_id

,comp\_asgloc\_country\_nm

,comp\_asgloc\_sens\_flg

,asgloc\_orgnbr

,asgloc\_id

,asgloc\_nm

,asgloc\_orgtype\_cd

,asgloc\_orgtype\_nm

,asgloc\_area\_orgnbr

,asgloc\_area\_nm

,asgloc\_country\_id

,asgloc\_country\_nm

,asgloc\_sens\_flg

,within\_msn\_orgnbr

,within\_msn\_asgloc\_id

,within\_msn\_nm

,within\_msn\_sens\_flg

,eccl\_org\_orgnbr

,eccl\_org\_nm

,eccl\_org\_sens\_flg

,temp\_comp\_id

,temp\_comp\_nm

,temp\_comp\_asgloc\_orgnbr

,temp\_comp\_asgloc\_id

,temp\_comp\_asgloc\_nm

,temp\_comp\_asgloc\_sens\_flg

,temp\_asgloc\_orgnbr

,temp\_asgloc\_id

,temp\_asgloc\_nm

,temp\_asgloc\_sens\_flg

,temp\_within\_msn\_orgnbr

,temp\_within\_msn\_asgloc\_id

,temp\_within\_msn\_nm

,temp\_within\_msn\_sens\_flg

,temp\_eccl\_org\_orgnbr

,temp\_eccl\_org\_nm

,temp\_eccl\_org\_sens\_flg

,fund\_type\_id

,last\_pmt\_dt

,blbl\_month\_yn

,crnt\_pmt\_num

,total\_pmts

,fund\_unit\_id

,fund\_unit\_nm

,fund\_unit\_sens\_flg

,fund\_unit\_stake\_id

,fund\_unit\_stake\_nm

,fund\_unit\_stake\_sens\_flg

,fund\_unit\_area\_orgnbr

,fund\_unit\_area\_nm

,fund\_unit\_country\_id

,fund\_unit\_country\_nm

,fund\_unit\_currency\_used

,fund\_unit\_currency\_cd

,fund\_unit\_currency\_nm

,fund\_unit\_exchange\_rt

,fund\_unit\_equal\_amt

,fund\_unit\_commit\_amt

,fund\_unit\_local\_amt

,fund\_unit\_us\_amt

,msn\_currency\_used

,msn\_currency\_cd

,msn\_currency\_nm

,msn\_local\_amt

,msn\_us\_amt

,msn\_exchange\_rt

,msn\_facs\_num

)

**SELECT**

tallywkly\_seq.nextval

,**TRUNC**(lv\_run\_date-7) **AS** sweep\_dt

,**TRUNC**(lv\_run\_date) **AS** created\_dt

,gt.msny\_id

,gt.legacy\_miss\_id

,gt.enamem\_id

,em.enamem\_homecountry\_id

,em.procstat\_id

,**RTRIM**(gt.last\_nm || ', ' || gt.first\_nm || ' ' || gt.middle\_nm) **AS** msny\_nm

,em.misstyp\_id

,em.term\_months

,**DECODE**(alc.comp\_housing\_id,3,'Y','N') **AS** live\_at\_home\_yn *-- fixed during Sr. Exp work*

,em.enamem\_mission\_start\_date **AS** mtc\_dt

,ca.asgmnt\_start\_date **AS** arriv\_dt

,em.enamem\_release\_date **AS** release\_dt

,em.enamem\_anniversary\_date **AS** anniv\_dt

,ca.call\_mtcs\_id **AS** mtc\_id

,ctom.orgshortname **AS** mtc\_cd

,tf.name **AS** mtc\_nm

,ca.call\_type **AS** call\_type\_id

,alc.comp\_id **AS** comp\_id

,alc.comp\_description **AS** comp\_nm

,cfom.orgnbr **AS** comp\_asgloc\_orgnbr

,alc.asgloc\_id **AS** comp\_asgloc\_id

,cal.name **AS** comp\_asgloc\_nm

,cfom.orgtypecode **AS** comp\_asgloc\_orgtype\_cd

,cfom.orgtypeshortdesctxt **AS** comp\_asgloc\_orgtype\_nm

,cpc.country\_poldivid **AS** comp\_asgloc\_country\_id

,cpc.country\_longname **AS** comp\_asgloc\_country\_nm

,cfom.orgsensitivitycode **AS** comp\_asgloc\_sens\_flg

,mfom.orgnbr **AS** asgloc\_orgnbr

,ca.call\_aloc\_id **AS** asgloc\_id

,mama.asgloc\_name **AS** asgloc\_nm

,mfom.orgtypecode **AS** asgloc\_orgtype\_cd

,mfom.orgtypeshortdesctxt **AS** asgloc\_orgtype\_nm

,mama.area\_orgnbr **AS** asgloc\_area\_orgnbr

,mama.area\_name **AS** asgloc\_area\_nm

,mpc.country\_poldivid **AS** asgloc\_country\_id

,mpc.country\_longname **AS** asgloc\_country\_nm

,mfom.orgsensitivitycode **AS** asgloc\_sens\_flg

,mama.wm\_orgnbr **AS** within\_msn\_orgnbr

,wmfom.asgloc\_id **AS** within\_msn\_asgloc\_id

,mama.wm\_orglongname **AS** within\_msn\_nm

,wmfom.orgsensitivitycode **AS** within\_msn\_sens\_flg

,mama.eccl\_orgnbr **AS** eccl\_org\_orgnbr

,mama.eccl\_orglongname **AS** eccl\_org\_nm

,efom.orgsensitivitycode **AS** eccl\_org\_sens\_flg

,cta.call\_comp\_id **AS** temp\_comp\_id

,talc.comp\_description **AS** temp\_comp\_nm

,tcfom.orgnbr **AS** temp\_comp\_asgloc\_orgnbr

,talc.asgloc\_id **AS** temp\_comp\_asgloc\_id

,tcal.name **AS** temp\_comp\_asgloc\_nm

,**NVL**(tcfom.orgsensitivitycode,0) **AS** temp\_comp\_asgloc\_sens\_flg

,tmfom.orgnbr **AS** temp\_asgloc\_orgnbr

,cta.call\_aloc\_id **AS** temp\_asgloc\_id

,tama.asgloc\_name **AS** temp\_asgloc\_nm

,**NVL**(tmfom.orgsensitivitycode,0) **AS** temp\_asgloc\_sens\_flg

,tama.wm\_orgnbr **AS** temp\_within\_msn\_orgnbr

,twmfom.asgloc\_id **AS** temp\_within\_msn\_asgloc\_id

,tama.wm\_orglongname **AS** temp\_within\_msn\_nm

,**NVL**(twmfom.orgsensitivitycode,0) **AS** temp\_within\_msn\_sens\_flg

,tama.eccl\_orgnbr **AS** temp\_eccl\_org\_orgnbr

,tama.eccl\_orglongname **AS** temp\_eccl\_org\_nm

,**NVL**(tefom.orgsensitivitycode,0) **AS** temp\_eccl\_org\_sens\_flg

,fmf.fund\_type\_id

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** lpm.last\_pay\_dt **ELSE** **NULL** **END** **AS** last\_pmt\_dt

,**CASE**

**WHEN** (lpm.last\_pay\_dt **is** **NULL** **OR** fmf.fund\_type\_id **IS** **NULL**)

**THEN** 'N' *-- not enough to go on*

**WHEN** (em.procstat\_id = 11 **AND** **TRUNC**(em.enamem\_mission\_start\_date, 'Month') > **TRUNC**(lv\_run\_date))

**THEN** 'N' *-- prefield that won't be in the MTC this month*

**WHEN** fmf.fund\_type\_id = 3 **OR** (**MONTHS\_BETWEEN**(**TRUNC**(lpm.last\_pay\_dt,'Month'),lv\_run\_date) < 0)

**THEN** 'N' *-- unsupported or those whose last payment month has passed*

**ELSE** 'Y'

**END** **AS** blbl\_month\_yn

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** **NVL**(gt.pmts\_sent,0)+1 **ELSE** **NULL** **END** **AS** crnt\_pmt\_num

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** **NVL**(gt.pmts\_to\_send,0) **ELSE** **NULL** **END** **AS** total\_pmts

,fmf.fund\_unit\_id **AS** fund\_unit\_id

,ffom.orglongname **AS** fund\_unit\_nm

,**NVL**(ffom.orgsensitivitycode,0) **AS** fund\_unit\_sens\_flg

,ffom.parentorgnbr **AS** fund\_unit\_stake\_orgnbr

,fstake.orglongname **AS** fund\_unit\_stake\_nm

,**NVL**(fstake.orgsensitivitycode,0) **AS** fund\_unit\_stake\_sens\_flg

,ffom.areaorgnbr **AS** fund\_unit\_area\_orgnbr

,farea.orglongname **AS** fund\_unit\_area\_nm

,ffpc.country\_poldivid **AS** fund\_unit\_country\_id

,ffpc.country\_longname **AS** fund\_unit\_country\_nm

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** ffp.cfis\_local\_or\_us\_flag **ELSE** **NULL** **END** **AS** fund\_unit\_currency\_used

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** fcurr.iso\_currency\_code **ELSE** **NULL** **END** **AS** fund\_unit\_currency\_cd

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** fcurr.currency\_name **ELSE** **NULL** **END** **AS** fund\_unit\_currency\_nm

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2,4,5) **THEN** fpcr.exch\_rate **ELSE** **NULL** **END** **AS** fund\_unit\_exchange\_rt

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN** ffp.finpol\_equalized\_pay\_amt

**WHEN** fmf.fund\_type\_id **IN** (4,5) **THEN** alc.housing\_exp

**ELSE** **NULL**

**END** **AS** fund\_unit\_equal\_amt

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN**

*-- add up commitment sources as reported by junior missionary*

**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0)

**WHEN** fmf.fund\_type\_id **IN** (4,5) **THEN**

*-- get housing commitment amount*

fmf.housing\_cmt\_amt

**END** **AS** fund\_unit\_commit\_amt

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN**

*-- Local Amount Due from Funding Unit = commitments if not equalized, otherwise the equalized amount*

*-- This formula for local and US amounts came from Legacy Pro\*C file sweeptally.pc*

**CASE**

*-- check unit override table for equalization flag, before using eq flag for country*

**WHEN** **NVL**(fuo.equalized\_yn, ffp.finpol\_equalized\_yn) = 'Y' **THEN**

ffp.finpol\_equalized\_pay\_amt

**ELSE**

**LEAST**(**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0),

**CASE**

**WHEN** (ffp.finpol\_equalized\_pay\_amt **IS** **NULL** **OR** ffp.finpol\_equalized\_pay\_amt = 0) **THEN**

**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0)

**ELSE**

ffp.finpol\_equalized\_pay\_amt

**END**

)

**END**

**WHEN** fmf.fund\_type\_id = 4 **THEN**

**CASE** **WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y' **AND** fmf.housing\_exp\_amt > 0) **THEN**

fmf.housing\_exp\_amt *-- missionary housing expense is stored in local currency of funding unit*

**ELSE** 0

**END**

**WHEN** fmf.fund\_type\_id = 5 **THEN**

**CASE**

**WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y') **THEN**

*-- Compare housing commitment to housing expense. Take the lesser.*

**CASE** **WHEN** fmf.housing\_cmt\_amt <= fmf.housing\_exp\_amt **THEN**

fmf.housing\_cmt\_amt

**WHEN** fmf.housing\_cmt\_amt > fmf.housing\_exp\_amt **THEN**

fmf.housing\_exp\_amt

**END**

**ELSE** 0

**END**

**ELSE** 0

**END** **AS** fund\_unit\_local\_amt

*-- USD amt due same as local, but multiplied by USD exchange rate*

,**CASE** **WHEN** fmf.fund\_type\_id **IN** (1,2) **THEN**

*-- USD Amount Due from Funding Unit = commitments if not equalized, otherwise the equalized amount*

*-- This formula for local and US amounts came from Legacy Pro\*C file sweeptally.pc*

**ROUND**(

**CASE**

*-- check unit override table for equalization flag, before using eq flag for country*

**WHEN** **NVL**(fuo.equalized\_yn, ffp.finpol\_equalized\_yn) = 'Y' **THEN**

ffp.finpol\_equalized\_pay\_amt

**ELSE**

**LEAST**(**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0),

**CASE**

**WHEN** (ffp.finpol\_equalized\_pay\_amt **IS** **NULL** **OR** ffp.finpol\_equalized\_pay\_amt = 0) **THEN**

**NVL**(fmf.fin\_self,0) + **NVL**(fmf.fin\_family,0) + **NVL**(fmf.fin\_ward,0) + **NVL**(fmf.fin\_other,0)

**ELSE**

ffp.finpol\_equalized\_pay\_amt

**END**

)

**END** \* fpcr.exch\_rate

)

**WHEN** fmf.fund\_type\_id = 4 **THEN**

**CASE** **WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y' **AND** fmf.housing\_exp\_amt > 0) **THEN**

**ROUND**(fmf.housing\_exp\_amt \* fpcr.exch\_rate)*-- missionary housing expense in terms of USD*

**ELSE** 0

**END**

**WHEN** fmf.fund\_type\_id = 5 **THEN**

**CASE**

**WHEN** (alc.sweep\_yn = 'Y' **AND** fmf.housing\_cap\_ppt\_yn = 'Y') **THEN**

*-- Compare commitment to housing expense and take the lesser*

**ROUND**(

**CASE** **WHEN** fmf.housing\_cmt\_amt <= fmf.housing\_exp\_amt **THEN**

fmf.housing\_cmt\_amt

**WHEN** fmf.housing\_cmt\_amt > fmf.housing\_exp\_amt **THEN**

fmf.housing\_exp\_amt

**END** \* fpcr.exch\_rate

)

**ELSE** 0

**END**

**ELSE** 0

**END** **AS** fund\_unit\_us\_amt

,mfp.cfis\_local\_or\_us\_flag **AS** msn\_currency\_used

,mcurr.iso\_currency\_code **AS** msn\_currency\_cd

,mcurr.currency\_name **AS** msn\_currency\_nm

*-- this formula for local and US amounts came from Legacy Pro\*C file sweeptally.pc*

,**CASE** **WHEN** mfp.cfis\_local\_or\_us\_flag = 'L' **THEN** mfa.base\_amt **ELSE** **ROUND**(mfa.base\_amt / mpcr.exch\_rate) **END** **AS** msn\_local\_amt

,**CASE** **WHEN** mfp.cfis\_local\_or\_us\_flag = 'L' **THEN** **ROUND**(mfa.base\_amt \* mpcr.exch\_rate) **ELSE** mfa.base\_amt **END** **AS** msn\_us\_amt

,mpcr.exch\_rate **AS** msn\_exchange\_rt

,mfa.gl\_unit\_num **AS** msn\_facs\_num

**FROM** msny\_week\_tally\_cand\_gt gt *-- this is the filter for what gets swept*

**JOIN** enabled\_members em **ON** em.enamem\_id = gt.enamem\_id

**JOIN** current\_perm\_assignment\_vw ca **ON** ca.call\_emem\_id = gt.enamem\_id

**JOIN** fin\_msny\_funding fmf **ON** fmf.enamem\_id = gt.enamem\_id

**JOIN** (

**SELECT** g.msny\_id,

**CASE**

*-- How many days beyond planned release is the actual release?*

**WHEN** (e.enamem\_release\_date **IS** **NULL** **OR** e.enamem\_anniversary\_date **IS** **NULL**) **THEN**

**NULL** *-- return empty value to be picked up by error module*

**WHEN** (e.enamem\_release\_date - e.enamem\_anniversary\_date) > 14 **THEN**

**ADD\_MONTHS**(e.enamem\_anniversary\_date,

(**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))

+(**CASE**

**WHEN** e.enamem\_release\_date - (**ADD\_MONTHS**(e.enamem\_anniversary\_date,**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))) > 14

**THEN** 0

**ELSE** -1

**END**)

)

**WHEN** (e.enamem\_release\_date - e.enamem\_anniversary\_date) < -14 **THEN**

**ADD\_MONTHS**(e.enamem\_anniversary\_date,

(**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))

-(**CASE**

**WHEN** e.enamem\_release\_date - (**ADD\_MONTHS**(e.enamem\_anniversary\_date,**TRUNC**(**MONTHS\_BETWEEN**(e.enamem\_release\_date,e.enamem\_anniversary\_date)))) < -14

**THEN** 2

**ELSE** 1

**END**)

)

**ELSE**

**ADD\_MONTHS**(e.enamem\_anniversary\_date,-1)

**END** **AS** last\_pay\_dt

**FROM** fin\_sweep\_msny\_gt g

**JOIN** enabled\_members e **ON** e.enamem\_id = g.enamem\_id

) lpm

**ON** lpm.msny\_id = gt.msny\_id

**JOIN** asg\_loc\_components alc **ON** alc.comp\_id = ca.call\_comp\_id

*-- component-level asgloc info*

**JOIN** fin\_organization\_mv cfom **ON** cfom.asgloc\_id = alc.asgloc\_id

**JOIN** assignment\_locations cal **ON** cal.asgloc\_id = alc.asgloc\_id

**JOIN** poldiv\_to\_country\_mv cpc **ON** cpc.primary\_poldivid = cfom.primarylocpoldivid

*-- "parent" asgloc info*

**JOIN** fin\_organization\_mv mfom **ON** mfom.asgloc\_id = ca.call\_aloc\_id

**JOIN** asgloc\_msn\_area\_mv mama **ON** mama.asgloc\_orgnbr = mfom.orgnbr

**JOIN** poldiv\_to\_country\_mv mpc **ON** mpc.primary\_poldivid = mfom.primarylocpoldivid

*-- "within mission" info*

**JOIN** fin\_organization\_mv wmfom **ON** wmfom.orgnbr = mama.wm\_orgnbr

*-- ecclesiastical info*

**JOIN** fin\_organization\_mv efom **ON** efom.orgnbr = mama.eccl\_orgnbr

*-- joins to get financial info for the mission*

**JOIN** fin\_poldivs mfp **ON** mfp.poldivid = mpc.country\_poldivid

**JOIN** mdm\_currency mcurr **ON** mcurr.iso\_currency\_code = mfp.finpol\_iso\_currency\_code

**JOIN** fin\_exch\_rate mpcr **ON** mpcr.from\_cur = mcurr.iso\_currency\_code

**JOIN** fin\_asglocs mfa **ON** mfa.asgloc\_id = mfom.asgloc\_id

*-- start of outer joins ---------------------------------------------*

**LEFT** **JOIN** training\_facilities tf **ON** tf.trnfac\_id = ca.call\_mtcs\_id *-- many MTC IDs are null, some 0*

**LEFT** **JOIN** col\_training\_orgs\_mv ctom **ON** ctom.orgnbr = tf.col\_orgnbr *-- must left join as well if MTC was null*

**LEFT** **JOIN** fin\_organization\_mv ffom **ON** ffom.orgnbr = fmf.fund\_unit\_id *-- some test missionaries won't have entries in fin\_organization\_mv*

**LEFT** **JOIN** fin\_organization\_mv fstake **ON** fstake.orgnbr = ffom.parentorgnbr *-- need long name of stake*

**LEFT** **JOIN** fin\_organization\_mv farea **ON** farea.orgnbr = ffom.areaorgnbr *-- need long name of area*

**LEFT** **JOIN** poldiv\_to\_country\_mv fpc **ON** fpc.primary\_poldivid = ffom.primarylocpoldivid

**LEFT** **JOIN** fin\_unit\_override fuo

**ON** fuo.orgnbr = **CASE**

**WHEN** fmf.allow\_override\_yn = 'Y' **THEN** ffom.orgnbr

**ELSE** **NULL** *-- remove ability to join to override table*

**END**

**LEFT** **JOIN** fin\_poldivs ffp *-- join for funding unit country financial attributes*

**ON** ffp.poldivid = **CASE**

**WHEN** fuo.orgnbr **IS** **NULL** **THEN** fpc.country\_poldivid

**ELSE** fuo.poldivid\_of\_equalized\_amt

**END**

*-- Now that fin\_poldivs is joined, perhaps on redirected country from poldivid\_of\_equalized\_amt,*

*-- re-join to poldiv\_to\_country\_mv to get country info.*

**LEFT** **JOIN** poldiv\_to\_country\_mv ffpc **ON** ffpc.primary\_poldivid = ffp.poldivid

**LEFT** **JOIN** mdm\_currency fcurr **ON** fcurr.iso\_currency\_code = ffp.finpol\_iso\_currency\_code

**LEFT** **JOIN** fin\_exch\_rate fpcr **ON** fpcr.from\_cur = fcurr.iso\_currency\_code

*-- Plethora of outer joins to get possible temp assignment info*

**LEFT** **JOIN** (

*-- current\_temp\_assignment\_vw didn't work since it internally used SYSDATE to filter. We*

*-- need the filter to use the sweep date, weekly*

**SELECT** call\_id, call\_aloc\_id, call\_comp\_id, call\_emem\_id, asgmnt\_start\_date, asgmnt\_end\_date

**FROM** napi\_assignment\_mstr

**WHERE** call\_type **IN** (2,3)

**AND** asgmnt\_start\_date <= lv\_run\_date

**AND** (asgmnt\_end\_date > lv\_run\_date **OR** asgmnt\_end\_date **IS** **NULL**)

**AND** status <> 7

) cta **ON** cta.call\_emem\_id = em.enamem\_id

**LEFT** **JOIN** asg\_loc\_components talc **ON** talc.comp\_id = cta.call\_comp\_id

**LEFT** **JOIN** fin\_organization\_mv tcfom **ON** tcfom.asgloc\_id = talc.asgloc\_id

**LEFT** **JOIN** assignment\_locations tcal **ON** tcal.asgloc\_id = talc.asgloc\_id

**LEFT** **JOIN** fin\_organization\_mv tmfom **ON** tmfom.asgloc\_id = cta.call\_aloc\_id

**LEFT** **JOIN** asgloc\_msn\_area\_mv tama **ON** tama.asgloc\_id = cta.call\_aloc\_id

**LEFT** **JOIN** fin\_organization\_mv twmfom **ON** twmfom.orgnbr = tama.wm\_orgnbr

**LEFT** **JOIN** fin\_organization\_mv tefom **ON** tefom.orgnbr = tama.eccl\_orgnbr

;

**COMMIT**;

*/\*\*\* store weekly grouped info for PERM serving missionaries \*\*\*/*

**INSERT** **INTO** msny\_weekly\_tally

( sweep\_dt, created\_dt

, comp\_id, comp\_nm

, CITIZENSHIP\_ID

, comp\_asgloc\_orgnbr, comp\_asgloc\_orgtype\_cd, comp\_asgloc\_orgtype\_nm

, comp\_asgloc\_id, comp\_asgloc\_nm, comp\_asgloc\_sens\_flg

, asgloc\_orgnbr, asgloc\_id, asgloc\_nm

, asgloc\_area\_orgnbr, asgloc\_area\_nm

, asgloc\_country\_id, asgloc\_country\_nm, asgloc\_sens\_flg

, within\_msn\_orgnbr, within\_msn\_asgloc\_id, within\_msn\_nm, within\_msn\_sens\_flg

, eccl\_org\_orgnbr, eccl\_org\_nm, eccl\_org\_sens\_flg

, msn\_currency\_used, msn\_currency\_cd, msn\_currency\_nm, msn\_exchange\_rt

, msn\_local\_amt, msn\_us\_amt, msn\_facs\_num

, pos\_spcl\_id, prim\_pos\_nm, prim\_pos\_pros\_only\_yn

, rspbl\_org\_nm, rspbl\_orgnbr

, lang\_id, lang\_nm

, complement\_elders, complement\_sisters, complement\_couples, complement\_selders, complement\_ssisters, prefield\_elders

, prefield\_sisters, prefield\_couples, prefield\_selders, prefield\_ssisters

, mtc\_elders, mtc\_sisters, mtc\_couples, mtc\_selders, mtc\_ssisters

, infield\_elders, infield\_sisters, infield\_couples, infield\_selders, infield\_ssisters

, temp\_elders, temp\_sisters, temp\_couples, temp\_selders, temp\_ssisters

, away\_elders, away\_sisters, away\_couples, away\_selders, away\_ssisters

, lah\_couples, lah\_selders, lah\_ssisters

)

**SELECT** \* **FROM** (

**WITH** finsm **AS**

(**SELECT** \* **FROM** msny\_week\_tally\_gt **WHERE** sweep\_dt = **TRUNC**(lv\_run\_date-7))

**SELECT**

sweep\_dt, lv\_run\_date

, comp\_id, comp\_nm

, citizenship\_id

, comp\_asgloc\_orgnbr, comp\_asgloc\_orgtype\_cd, comp\_asgloc\_orgtype\_nm

, comp\_asgloc\_id, comp\_asgloc\_nm, comp\_asgloc\_sens\_flg

, asgloc\_orgnbr , asgloc\_id, asgloc\_nm

, asgloc\_area\_orgnbr, asgloc\_area\_nm

, asgloc\_country\_id, asgloc\_country\_nm, asgloc\_sens\_flg

, within\_msn\_orgnbr, within\_msn\_asgloc\_id, within\_msn\_nm, within\_msn\_sens\_flg

, eccl\_org\_orgnbr, eccl\_org\_nm, eccl\_org\_sens\_flg

, msn\_currency\_used, msn\_currency\_cd, msn\_currency\_nm, msn\_exchange\_rt

, msn\_local\_amt, msn\_us\_amt, msn\_facs\_num

, pos\_spcl\_id, prim\_pos\_nm, prim\_pos\_pros\_only\_yn

, rspbl\_org\_nm, rspbl\_orgnbr

, lang\_id, lang\_nm

, complement\_elders, complement\_sisters, complement\_couples, complement\_selders, complement\_ssisters

, prefield\_elders, prefield\_sisters, prefield\_couples, prefield\_selders, prefield\_ssisters

, mtc\_elders, mtc\_sisters, mtc\_couples, mtc\_selders, mtc\_ssisters

, infield\_elders, infield\_sisters, infield\_couples, infield\_selders, infield\_ssisters

, **CASE** **WHEN** misstyp\_id = 1 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_elders

, **CASE** **WHEN** misstyp\_id = 2 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_sisters

, **CASE** **WHEN** misstyp\_id = 3 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_couples

, **CASE** **WHEN** misstyp\_id = 4 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_selders

, **CASE** **WHEN** misstyp\_id = 5 **THEN** **NVL**(tfsw.msny\_count,0) **ELSE** 0 **END** **AS** temp\_ssisters

, away\_elders, away\_sisters, away\_couples, away\_selders, away\_ssisters

, lah\_couples, lah\_selders, lah\_ssisters

**FROM**

( **SELECT**

fsm.sweep\_dt

, fsm.comp\_id, fsm.comp\_nm

, fsm.citizenship\_id

, fsm.asgloc\_id

, fsm.asgloc\_nm

, fsm.fund\_unit\_id, fsm.comp\_asgloc\_orgnbr

, fsm.comp\_asgloc\_orgtype\_cd

, fsm.comp\_asgloc\_orgtype\_nm

, fsm.comp\_asgloc\_sens\_flg

, fsm.comp\_asgloc\_id

, fsm.comp\_asgloc\_nm

, fsm.asgloc\_orgnbr

, fsm.asgloc\_area\_orgnbr

, fsm.asgloc\_area\_nm

, fsm.asgloc\_country\_id

, fsm.asgloc\_country\_nm

, fsm.asgloc\_sens\_flg

, fsm.within\_msn\_orgnbr

, fsm.within\_msn\_asgloc\_id

, fsm.within\_msn\_nm

, fsm.within\_msn\_sens\_flg

, fsm.eccl\_org\_orgnbr

, fsm.eccl\_org\_nm

, fsm.eccl\_org\_sens\_flg

, fsm.msn\_currency\_used

, fsm.msn\_currency\_cd

, fsm.msn\_currency\_nm

, fsm.msn\_exchange\_rt

, fsm.msn\_local\_amt

, fsm.msn\_us\_amt

, fsm.msn\_facs\_num

, acpv.posspc\_id pos\_spcl\_id

, acpv.primary\_pos\_name prim\_pos\_nm

, acpv.primary\_pos\_pros\_only prim\_pos\_pros\_only\_yn

, acpv.responsible\_org rspbl\_org\_nm

, acpv.responsible\_orgnbr rspbl\_orgnbr

, acpv.lang\_id

, acpv.language\_name lang\_nm

, alc.misstyp\_id

, **CASE** **WHEN** alc.misstyp\_id = 1 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_elders

, **CASE** **WHEN** alc.misstyp\_id = 2 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_sisters

, **CASE** **WHEN** alc.misstyp\_id = 3 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_couples

, **CASE** **WHEN** alc.misstyp\_id = 4 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_selders

, **CASE** **WHEN** alc.misstyp\_id = 5 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 11 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** prefield\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 12 **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** mtc\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NULL** **OR** fsm.temp\_comp\_id = fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** infield\_ssisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 1 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_elders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 2 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_sisters

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 3 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_couples

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 4 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_selders

, **SUM**(**CASE** **WHEN** alc.misstyp\_id = 5 **THEN** **CASE** **WHEN** fsm.procstat\_id = 13 **AND** (fsm.temp\_asgloc\_id **IS** **NOT** **NULL** **AND** fsm.temp\_comp\_id <> fsm.comp\_id) **THEN** 1 **ELSE** 0 **END** **ELSE** 0 **END**) **AS** away\_ssisters

, **SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** alc.misstyp\_id = 3 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_couples

, **SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** alc.misstyp\_id = 4 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_selders

, **SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** alc.misstyp\_id = 5 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_ssisters

**FROM** (**SELECT** \* **FROM** msny\_week\_tally\_gt **WHERE** sweep\_dt = **TRUNC**(lv\_run\_date-7)) fsm

**JOIN** asg\_loc\_components alc **ON** alc.comp\_id = fsm.comp\_id

**JOIN** asgloc\_comp\_pos\_vw acpv **ON** acpv.comp\_id = fsm.comp\_id *-- join for position info*

**GROUP** **BY**

fsm.sweep\_dt

, fsm.comp\_id

, fsm.comp\_nm

, fsm.citizenship\_id

, fsm.asgloc\_id

, fsm.asgloc\_nm

, fsm.fund\_unit\_id

, fsm.comp\_asgloc\_id

, fsm.comp\_asgloc\_nm

, fsm.comp\_asgloc\_orgnbr, fsm.comp\_asgloc\_orgtype\_cd, fsm.comp\_asgloc\_orgtype\_nm

, fsm.comp\_asgloc\_sens\_flg

, fsm.asgloc\_orgnbr

, fsm.asgloc\_area\_orgnbr, fsm.asgloc\_area\_nm

, fsm.asgloc\_country\_id, fsm.asgloc\_country\_nm, fsm.asgloc\_sens\_flg

, fsm.within\_msn\_orgnbr, fsm.within\_msn\_asgloc\_id, fsm.within\_msn\_nm, fsm.within\_msn\_sens\_flg

, fsm.eccl\_org\_orgnbr, fsm.eccl\_org\_nm, fsm.eccl\_org\_sens\_flg

, fsm.msn\_currency\_used, fsm.msn\_currency\_cd, fsm.msn\_currency\_nm, fsm.msn\_exchange\_rt

, fsm.msn\_local\_amt, fsm.msn\_us\_amt, fsm.msn\_facs\_num

, acpv.posspc\_id, acpv.primary\_pos\_name, acpv.primary\_pos\_pros\_only

, acpv.responsible\_org, acpv.responsible\_orgnbr

, acpv.lang\_id, acpv.language\_name

, alc.comp\_num

, alc.misstyp\_id

) fsm

**LEFT** **OUTER** **JOIN** (**SELECT** temp\_comp\_id, **COUNT**(\*) msny\_count **FROM** finsm **GROUP** **BY** temp\_comp\_id) tfsw **ON** tfsw.temp\_comp\_id = fsm.comp\_id) ;

**COMMIT**;

*/\*\*\* store weekly grouped info for TEMP serving missionaries \*\*\*/*

**INSERT** **INTO** msny\_weekly\_tally

( sweep\_dt, created\_dt

, comp\_id, comp\_nm

, CITIZENSHIP\_ID

, comp\_asgloc\_orgnbr, comp\_asgloc\_orgtype\_cd, comp\_asgloc\_orgtype\_nm

, comp\_asgloc\_id, comp\_asgloc\_nm, comp\_asgloc\_sens\_flg

, asgloc\_orgnbr, asgloc\_id, asgloc\_nm

, asgloc\_area\_orgnbr, asgloc\_area\_nm

, asgloc\_country\_id, asgloc\_country\_nm, asgloc\_sens\_flg

, within\_msn\_orgnbr, within\_msn\_asgloc\_id, within\_msn\_nm, within\_msn\_sens\_flg

, eccl\_org\_orgnbr, eccl\_org\_nm, eccl\_org\_sens\_flg

, msn\_currency\_used, msn\_currency\_cd, msn\_currency\_nm, msn\_exchange\_rt

, msn\_local\_amt, msn\_us\_amt, msn\_facs\_num

, pos\_spcl\_id, prim\_pos\_nm, prim\_pos\_pros\_only\_yn

, rspbl\_org\_nm, rspbl\_orgnbr

, lang\_id, lang\_nm

, complement\_elders, complement\_sisters, complement\_couples, complement\_selders, complement\_ssisters, prefield\_elders

, prefield\_sisters, prefield\_couples, prefield\_selders, prefield\_ssisters

, mtc\_elders, mtc\_sisters, mtc\_couples, mtc\_selders, mtc\_ssisters

, infield\_elders, infield\_sisters, infield\_couples, infield\_selders, infield\_ssisters

, temp\_elders, temp\_sisters, temp\_couples, temp\_selders, temp\_ssisters

, away\_elders, away\_sisters, away\_couples, away\_selders, away\_ssisters

, lah\_couples, lah\_selders, lah\_ssisters

)

**SELECT** \* **FROM**

(

**WITH** ot **AS** *-- "ot" stands for orphaned temporaries*

(**SELECT** **TRUNC**(lv\_run\_date-7)sw\_dt, **TRUNC**(lv\_run\_date) cr\_dt, fsm.temp\_comp\_id, fsm.temp\_comp\_nm,

citizenship\_id,

fsm.temp\_comp\_asgloc\_orgnbr, fsm.temp\_comp\_asgloc\_id, fsm.temp\_comp\_asgloc\_nm, fsm.temp\_comp\_asgloc\_sens\_flg,

fsm.temp\_asgloc\_orgnbr, fsm.temp\_asgloc\_id, fsm.temp\_asgloc\_nm, fsm.temp\_asgloc\_sens\_flg,

fsm.temp\_within\_msn\_orgnbr, fsm.temp\_within\_msn\_asgloc\_id, fsm.temp\_within\_msn\_nm, fsm.temp\_within\_msn\_sens\_flg,

fsm.temp\_eccl\_org\_orgnbr, fsm.temp\_eccl\_org\_nm, fsm.temp\_eccl\_org\_sens\_flg,

**SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** fsm.misstyp\_id = 3 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_couples,

**SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** fsm.misstyp\_id = 4 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_selders,

**SUM**(**CASE** **WHEN** fsm.live\_at\_home\_yn = 'Y' **AND** fsm.misstyp\_id = 5 **THEN** 1 **ELSE** 0 **END**) **AS** lah\_ssisters,

**COUNT**(\*) **AS** cnt

**FROM** msny\_week\_tally\_gt fsm

**WHERE** fsm.sweep\_dt = **TRUNC**(lv\_run\_date-7)

**AND** fsm.temp\_comp\_id **IS** **NOT** **NULL**

**AND** **NOT** **EXISTS** (**SELECT** **NULL**

**FROM** msny\_weekly\_tally fst

**WHERE** fst.sweep\_dt = **TRUNC**(lv\_run\_date-7)

**AND** fst.comp\_id = fsm.temp\_comp\_id)

**GROUP** **BY**

fsm.temp\_comp\_id, fsm.temp\_comp\_nm,

citizenship\_id,

fsm.temp\_comp\_asgloc\_orgnbr, fsm.temp\_comp\_asgloc\_id, fsm.temp\_comp\_asgloc\_nm, fsm.temp\_comp\_asgloc\_sens\_flg,

fsm.temp\_asgloc\_orgnbr, fsm.temp\_asgloc\_id, fsm.temp\_asgloc\_nm, fsm.temp\_asgloc\_sens\_flg,

fsm.temp\_within\_msn\_orgnbr, fsm.temp\_within\_msn\_asgloc\_id, fsm.temp\_within\_msn\_nm, fsm.temp\_within\_msn\_sens\_flg,

fsm.temp\_eccl\_org\_orgnbr, fsm.temp\_eccl\_org\_nm, fsm.temp\_eccl\_org\_sens\_flg

)

**SELECT**

**TRUNC**(lv\_run\_date-7) sw\_dt

, **TRUNC**(lv\_run\_date) cr\_dt

, ot.temp\_comp\_id **AS** comp\_id

, ot.temp\_comp\_nm **AS** comp\_nm

, citizenship\_id

, ot.temp\_comp\_asgloc\_orgnbr **AS** comp\_asgloc\_orgnbr, cfom.orgtypecode **AS** comp\_asgloc\_orgtype\_cd, cfom.orgtypeshortdesctxt **AS** comp\_asgloc\_orgtype\_nm

, ot.temp\_comp\_asgloc\_id **AS** comp\_asgloc\_id, ot.temp\_comp\_asgloc\_nm **AS** comp\_asgloc\_nm, ot.temp\_comp\_asgloc\_sens\_flg **AS** comp\_asgloc\_sens\_flg

, ot.temp\_asgloc\_orgnbr **AS** asgloc\_orgnbr, ot.temp\_asgloc\_id **AS** asgloc\_id, ot.temp\_asgloc\_nm **AS** asgloc\_nm

, acom.areaorgnbr **AS** asgloc\_area\_orgnbr, acom.areamediumname **AS** asgloc\_area\_nm

, apc.country\_poldivid asgloc\_country\_id, apc.country\_longname **AS** asgloc\_country\_nm, ot.temp\_asgloc\_sens\_flg **AS** asgloc\_sens\_flg

, ot.temp\_within\_msn\_orgnbr **AS** within\_msn\_orgnbr, ot.temp\_within\_msn\_asgloc\_id **AS** within\_msn\_asgloc\_id, ot.temp\_within\_msn\_nm **AS** within\_msn\_nm, ot.temp\_within\_msn\_sens\_flg **AS** within\_msn\_sens\_flg

, ot.temp\_eccl\_org\_orgnbr **AS** eccl\_org\_orgnbr, ot.temp\_eccl\_org\_nm **AS** eccl\_org\_nm, ot.temp\_eccl\_org\_sens\_flg **AS** eccl\_org\_sens\_flg

, afp.cfis\_local\_or\_us\_flag **AS** msn\_currency\_used, afp.finpol\_iso\_currency\_code **AS** msn\_currency\_cd

, acurr.currency\_desc **AS** msn\_currency\_nm, apcr.exch\_rate **AS** msn\_exchange\_rt

, **CASE** **WHEN** afp.cfis\_local\_or\_us\_flag = 'L' **THEN** afa.base\_amt **ELSE** **ROUND**(afa.base\_amt / apcr.exch\_rate) **END** **AS** msn\_local\_amt

, **CASE** **WHEN** afp.cfis\_local\_or\_us\_flag = 'L' **THEN** **ROUND**(afa.base\_amt \* apcr.exch\_rate) **ELSE** afa.base\_amt **END** **AS** msn\_us\_amt

, afa.gl\_unit\_num **AS** msn\_facs\_num

, acpv.posspc\_id pos\_spcl\_id, acpv.primary\_pos\_name prim\_pos\_nm, acpv.primary\_pos\_pros\_only prim\_pos\_pros\_only\_yn

, acpv.responsible\_org rspbl\_org\_nm, acpv.responsible\_orgnbr rspbl\_orgnbr

, acpv.lang\_id, acpv.language\_name lang\_nm

, **CASE** **WHEN** alc.misstyp\_id = 1 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_elders

, **CASE** **WHEN** alc.misstyp\_id = 2 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_sisters

, **CASE** **WHEN** alc.misstyp\_id = 3 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_couples

, **CASE** **WHEN** alc.misstyp\_id = 4 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_selders

, **CASE** **WHEN** alc.misstyp\_id = 5 **THEN** alc.comp\_num **ELSE** 0 **END** **AS** complement\_ssisters

*-- Missionaries with active temp assignments will always be infield. They can have planned*

*-- temp assignments while prefield or MTC, but those assignments won't show up*

*-- in current temp assignment inline view, which is used in the initial insert to fin\_sweep\_msny*

, 0 **AS** prefield\_elders

, 0 **AS** prefield\_sisters

, 0 **AS** prefield\_couples

, 0 **AS** prefield\_selders

, 0 **AS** prefield\_ssisters

, 0 **AS** mtc\_elders

, 0 **AS** mtc\_sisters

, 0 **AS** mtc\_couples

, 0 **AS** mtc\_selders

, 0 **AS** mtc\_ssisters

, 0 **AS** infield\_elders

, 0 **AS** infield\_sisters

, 0 **AS** infield\_couples

, 0 **AS** infield\_selders

, 0 **AS** infield\_ssisters

, **CASE** **WHEN** alc.misstyp\_id = 1 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_elders

, **CASE** **WHEN** alc.misstyp\_id = 2 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_sisters

, **CASE** **WHEN** alc.misstyp\_id = 3 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_couples

, **CASE** **WHEN** alc.misstyp\_id = 4 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_selders

, **CASE** **WHEN** alc.misstyp\_id = 5 **THEN** ot.cnt **ELSE** 0 **END** **AS** temp\_ssisters

, 0 **AS** away\_elders

, 0 **AS** away\_sisters

, 0 **AS** away\_couples

, 0 **AS** away\_selders

, 0 **AS** away\_ssisters

, ot.lah\_couples

, ot.lah\_selders

, ot.lah\_ssisters

**FROM** ot

**JOIN** assignment\_locations al

**ON** al.asgloc\_id = ot.temp\_asgloc\_id

**JOIN** asg\_loc\_components alc

**ON** alc.comp\_id = ot.temp\_comp\_id

**JOIN** assignment\_locations cal

**ON** cal.asgloc\_id = alc.asgloc\_id

**JOIN** fin\_organization\_mv cfom *-- join for component-level assignment location info*

**ON** cfom.orgnbr = cal.asgloc\_col\_orgnbr

**JOIN** asgloc\_comp\_pos\_vw acpv *-- join for position info*

**ON** acpv.comp\_id = ot.temp\_comp\_id

**JOIN** fin\_organization\_mv acom *-- join for assignment location info*

**ON** acom.orgnbr = al.asgloc\_col\_orgnbr

**JOIN** poldiv\_to\_country\_mv apc *-- join for assignment location country*

**ON** apc.primary\_poldivid = acom.primarylocpoldivid

**JOIN** fin\_poldivs afp *-- join for assignment location country financial attributes*

**ON** afp.poldivid = apc.country\_poldivid

**JOIN** currency\_type\_vw acurr

**ON** acurr.iso\_currency\_code = afp.finpol\_iso\_currency\_code

**JOIN** fin\_exch\_rate apcr

**ON** apcr.from\_cur = acurr.iso\_currency\_code

**JOIN** fin\_asglocs afa

**ON** afa.asgloc\_id = al.asgloc\_id

);

**COMMIT**;

**END** weekly\_tally;

**END** sweeps;