

```

/*****
*   PRODUCT:   SAS
*   VERSION:   6.12
*   CREATOR:   External File Interface - Version 2
*   DATE:      31Mar01
*   DESC:      Generated SAS Datastep Code
*   NAME:      Mainload_Allmonths.sas
*****/
/*Variables given values in Run_mainload_&_reports.sas */

%let campcode   = 1121; /* CampCode */
%let e_ext      = 0107; /* YearMonth*/
%let esisext    = 071;  /* CurrentMonth*/
%let esislst    = 061;  /* PreviousMonth*/
%let esisjune   = 061;  /* June data for urgency date comparisons
                          Added 1 July 2001 - DO NOT CHANGE */
%let CDATE = 31/7/01;    /* Default value*/
%let MONTH = 7;         /* Default value*/

%include "f:\data\esis_data\ref_slas\formats_for_sla_allocation.sas";
libname esis_iv "f:\data\esis_data\esis_iv\edata\";
libname esis_iii "F:\data\esis_data\esis iii\edata\";

%macro setlib;
  %if &esislst ne 061 %then
    libname esislst "f:\data\esis_data\esis_iv\edata\";
  %else libname esislst "f:\data\esis_data\esis iii\edata\";
%mend;
%setlib;

/*   Read control file to get the header record - cdate, campus code and   */
/*   num of records                                                         */

data esis_iv.con1;
  %let _EFIERR_ = 0 ; /* clear ERROR detection macro variable */
  infile
  "f:\data\esis_data\esis_iv\rawdata\0002\&campcode.\C&esisext.&campcode..TXT"
  delimiter = ',' MISSOVER DSD;
  /*infile "C:\data\esis data\C&esisext.&campcode..TXT" delimiter = ',' MISSOVER
  DSD obs=1;*/
  format cdate date9. ;
  format campcode $4. ;
  format n_rec best12. ;
  informat cdate ddmmyy10. ;
  informat campcode $4. ;
  informat n_rec best32. ;
  input
    cdate
    campcode $
    n_rec;

  If _ERROR_ then /* ERROR detection */
    call symput('_EFIERR_',1);
run;

data esis_iv.con2;
  set esis_iv.con1;

  format e_ext $4. rcdate date9.;

  e_ext=&e_ext.;

```

```

/*      This is required to check the correct census date      */

if e_ext = 0107 then rcdate = '31Jul2001'd;
if e_ext = 0108 then rcdate = '31Aug2001'd;
if e_ext = 0109 then rcdate = '30SEP2001'd;
if e_ext = 0110 then rcdate = '31OCT2001'd;
if e_ext = 0111 then rcdate = '30NOV2001'd;
if e_ext = 0112 then rcdate = '31DEC2001'd;
if e_ext = 0201 then rcdate = '31JAN2002'd;
if e_ext = 0202 then rcdate = '28FEB2002'd;
if e_ext = 0203 then rcdate = '31MAR2002'd;
if e_ext = 0204 then rcdate = '30APR2002'd;
if e_ext = 0205 then rcdate = '31MAY2002'd;
if e_ext = 0206 then rcdate = '30JUN2002'd;

if e_ext = 0107 then call symput('CDATE','31/07/2001');
if e_ext = 0108 then call symput('CDATE','31/08/2001');
if e_ext = 0109 then call symput('CDATE','30/09/2001');
if e_ext = 0110 then call symput('CDATE','31/10/2001');
if e_ext = 0111 then call symput('CDATE','30/11/2001');
if e_ext = 0112 then call symput('CDATE','31/12/2001');
if e_ext = 0201 then call symput('CDATE','31/01/2002');
if e_ext = 0202 then call symput('CDATE','28/02/2002');
if e_ext = 0203 then call symput('CDATE','31/03/2002');
if e_ext = 0204 then call symput('CDATE','30/04/2002');
if e_ext = 0205 then call symput('CDATE','31/05/2002');
if e_ext = 0206 then call symput('CDATE','30/06/2002');

run;

/***** READ CURRENT DATA FILE TO GET PATIENT RECORD *****/

data esis_iv.d&esisext.;
  %let _EFIERR_ = 0;      /* clear ERROR detection macro variable */
  infile
  "f:\data\esis_data\esis_iv\rawdata\0002\&campcode.\D&esisext.&campcode..TXT"
  delimiter = ',' MISSOVER DSD;

  format key $varying8.      ;      /* 1 */
  format urno $varying10.   ;      /* 2 */
  format mednum $11.        ;      /* 3 */
  format medsuf $upcase3.   ;      /* 4 */
  format waitno $2.         ;      /* 5 */
  format birthdat date9.    ;      /* 6 */
  format sex $1.           ;      /* 7 */
  format pstcode $4.        ;      /* 8 */
  format locality $30.      ;      /* 9 */
  format ppproc z3.         ;      /* 10 writes leading zeros to length of
3*/
  format surgspec z2.       ;      /* 11 writes leading zeros to length of
2*/

```

```

format urgency $1. ; /* 12*/
format planstay $1. ; /* 13*/
format reg_date date9. ; /* 14*/
format refsourc $1. ; /* 15*/
format refhosp $4. ; /* 16*/
format urg_date date9. ; /* 17*/
format urgup_dt date9. ; /* 18 */
format old_urg $1. ; /* 19*/
format pat_stat $1. ; /* 20*/
format sr_date date9. ; /* 21*/
format nrreason $1. ; /* 22*/
format bookdate date9. ; /* 23*/
format bookno z2.0 ; /* 24 add leading zero to make
consistent with hi_post*/
format rbreason $1. ; /* 25*/
format sch_date date9. ; /* 26*/
format rem_date date9. ; /* 27*/
format rmreason $1. ; /* 28*/
format insure $1. ; /* 29*/
format transdes $4. ; /* 30*/
format nr_total z4.0 ; /* 31 writes leading zeros to length of
4*/
format nr_assig z4. ; /* 32 writes leading zeros to length of
4*/
format nrtot_ui z4. ; /* 33 writes leading zeros to length of
4*/
format hi_post z2.0 ; /* 34 add leading zero to make
consistent with bookno*/
informat key $varying8. ;
informat urno $varying12. ;
informat mednum $11. ;
informat medsuf $3. ;
informat waitno $2. ;
informat birthdat ddmmyy10. ;
informat sex $1. ;
informat pstcode $4. ;
informat locality $30. ;
informat ppproc 3.0 ;
informat surgspec 2.0 ;
informat urgency $1. ;
informat planstay $1. ;
informat reg_date ddmmyy10. ;
informat refsourc $1. ;
informat refhosp $4. ;
informat urg_date ddmmyy10. ;
informat urgup_dt ddmmyy10. ;
informat old_urg $1. ;
informat pat_stat $1. ;
informat sr_date ddmmyy10. ;
informat nrreason $1. ;
informat bookdate ddmmyy10. ;
informat bookno 2.0 ;
informat rbreason $1. ;
informat sch_date ddmmyy10. ;
informat rem_date ddmmyy10. ;

```

```

informat rmreason $1. ;
informat insure $1. ;
informat transdes $4. ;
informat nr_total 4.0 ;
informat nr_assig 4. ;
informat nrtot_ui 4. ;
informat hi_post 2.0 ;
input

```

```

    key $
    urno $
    mednum $
    medsuf $
    waitno $
    birthdat $
    sex $
    pstcode $
    locality $
    ppproc
    surgspec $
    urgency $
    planstay $
    reg_date
    refsourc $
    refhosp $
    urg_date
    urgup_dt $
    old_urg $
    pat_stat $
    sr_date
    nrreason $
    bookdate
    bookno
    rbreason $
    sch_date
    rem_date
    rmreason $
    insure $
    transdes $
    nr_total
    nr_assig
    nrtot_ui
    hi_post;

```

```

If _ERROR_ then          /* ERROR detection */
    call symput('_EFIERR_',1);

```

```
run;
```

```
/****** Merge in cdate date and campcode and num records with the data file*****/
```

```

data esis_iv.d&esisext.;
    if _n_=1 then set esis_iv.con2;
    set esis_iv.d&esisext.;
run;

```

```
/****** End Merge *****/
```

```

data esis_iv.e&esisext.;

    set esis_iv.d&esisext.;

```

```

/***** PROCESS B: CREATE DERIVED DATA FIELDS *****/

  epid = trimn(campcode)||trimn(urno)||trimn(waitno) ;    /* make epid */
run;

/***** To count for number of records in the data file *****/

data esis_iv.e&esisext.;
  set esis_iv.e&esisext.;
  count = 1;
run;

proc summary data=esis_iv.e&esisext. nway missing;
  class campcode;
  var count;
  output out=tcm (drop=_type_ _freq_)  n(count)=d_n_rec;
run;

data keep4later;
set tcm;
run;

data esis_iv.e&esisext.;
  if _n_ = 1 then set tcm;
  set esis_iv.e&esisext.;
run;

/*      End Record Count                                */

/*      To check if a different waitno exists with the same      */
/*      Patient Identifier and the same ppproc                    */

proc summary data=esis_iv.e&esisext. nway missing;
  class campcode urno ppproc;
  var count;
  output out=tcm1 (drop=_type_ _freq_)  n(count)=ppp_dup;
run;

proc sort data = esis_iv.e&esisext.;
  by urno ppproc;

proc sort data = work.tcm1 out = work.tcm1;
  by urno ppproc;

data esis_iv.e&esisext.;
  merge esis_iv.e&esisext.(in=ina) work.tcm1 (in=inb);
  by urno ppproc;
  if ina then output;
run;

```

```

/***** End ppp check *****/
%macro previous;
  %if &e_ext ne 0107 %then %do;
/***** READ PREVIOUS MONTH DATA FILE FOR MONTH TO MONTH COMPARISON *****/
  data e_old(keep=key urno waitno sexold birthold procold regold urgold plansold
wstatold
          booknold hiposold schold nrtotold nrassold nruiold oldurgdt
specold dset);
    set esislast.e&esislast.&campcode.(drop = birthold sexold procold regold
          specold urgold plansold wstatold hiposold schold oldurgdt
          nrtotold nrassold nruiold dset booknold);

    format ppproc 4. bookno 2. surgspec 2. hi_post 2.;
    format nr_total 4. nr_assig 4. nrtot_ui 4.;
    format nruiold 4. booknold 2. specold 2. procold 3. nrtotold 4.;
    format nrassold 4. hiposold 2. ;

    rename sex      = sexold;
    rename birthdat = birthold;
    rename ppproc   = procold;
    rename reg_date = regold;
    rename surgspec = specold;
    rename urgency  = urgold;
    rename planstay = plansold;
    rename waitstat = wstatold;
    rename bookno   = booknold;
    rename hi_post  = hiposold;
    rename sch_date = schold;
    rename urg_date = oldurgdt;
    if nr_total = . then nr_total = 0;
    if nr_assig = . then nr_assig = 0;
    if nrtot_ui = . then nrtot_ui = 0;
    rename nr_total = nrtotold;
    rename nr_assig = nrassold;
    rename nrtot_ui = nruiold;
    dset = 'old';
    if rmreason ne 'E';
run;
%end;
%else %do;
data e_old(keep=key urno waitno sexold birthold procold regold urgold plansold
wstatold
          booknold hiposold schold nrtotold nrassold nruiold oldurgdt
specold dset);
    set esislast.e&esislast.&campcode.;

    format ppproc 4. bookno 2. surgspec 2. hi_post 2.;
    format nr_total 4. nr_assig 4.;
    format nruiold 4. nrtotold 4.;
    format nrassold 4.;

```

```

rename sex      = sexold;
rename birthdat = birthold;
rename ppproc   = procold;
rename reg_date = regold;
rename surgspec = specold;
rename urgency = urgold;
rename planstay = plansold;
rename waitstat = wstatold;
rename bookno   = booknold;
rename hi_post  = hiposold;
rename sch_date = schold;
rename urg_date = oldurgdt;
nrtotold = 0;
nrassold = 0;
nruiold = 0;
dset = 'old';
if rmreason ne 'E';
run;
%end;
%mend;

%previous;
proc sort data=e_old;
  by key urno waitno sexold birthold;
run;

/***** END E_OLD *****/

data e_new;
  set esis_iv.e&esisext.;
  dset = 'new';
run;

proc sort data=e_new;
  by key urno waitno sex birthdat;
run;

data e_comb;
  merge e_old e_new;
  by key;
run;

/*** READ JUNE DATA FILE FOR URGENCY REASSIGNMENT DATE COMPARISONS
      Added 1 July 2001                                     *** /

data e_june(keep=key urno waitno Junurgdt Junnrass);
  set esis_iii.e&esisjune.&campcode.;
  keep key urno waitno urg_date nr_assig;
  if nr_assig = . then nr_assig = 0;
  rename urg_date = Junurgdt;
  rename nr_assig = Junnrass;
  if rmreason ne 'E';
run;

proc sort data=e_june;
  by key urno waitno;
run;

proc sort data=e_comb;
  by key urno waitno;
run;

```

```

data e_comp;
  merge e_comb e_june;
  by key;
run;

/*****
/***** PROCESS C: DATA FILE EDITS *****/
/*****

data esis_iv.z&esisext. esis_iv.r&esisext.&campcode.;

  set e_comp ;
  format type type1 type2 type3 8.0;
  type1 = 0; type2 = 0; type3 = 0; type = 0; ok = 0;
  format row_number 5.0;
  row_number = _n_;
  array mess{20} $80. ;          /* set up array for error mess          */

/* Blanket statement to ignore data errors from editing          */
  if rmreason ne 'E';
  if dset eq 'new' then do; /* CHECKS AGAINST NEW DATA ONLY */
/*****
/* S001: CENSUS DATE NOT VALID          */
/*****
  if (cdate ne rcdate) and dset ne 'old' then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type1:S001 Census Date Not Valid';
  end;

/*****
/* S002: NO HEADER RECORD FOUND          */
/*****

  if ((cdate = .) or (campcode = '') or (n_rec = .)) and dset ne 'old' then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type1:S002 No Header Record Found';
  end;

/*****
/* S003: INCORRECT CAMPUS CODE          */
/*****

  if ((campcode = '' ) or
      (campcode not in ('1010','1590','1031','2010','2050','1020',
                       '1050','2111','2220','1121','1210','1170',
'2440','1280','1191','1331','1240','1230',
'1360','1450','1390','1150','1580','1180'))
      and dset ne 'old' then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type1:S003 Incorrect Campus Code';
  end;

```

```

/*****
/* S004: TOTAL NUMBER OF RECORDS NOT EQUAL TO TOTAL PATIENT RECORDS      */
/*****

  if n_rec ne d_n_rec then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S004 Total Number Of Records Not Equal To Total Patient
Records';
  end;

/*****
/* S005: CONTROL FILENAME INCORRECT                                       */
/* Submission officer to manually check filename                           */
/* and inform the hospital of this error                                   */
/*****

/*****
/* S006: CONTROL FILE CONTENT INCORRECT                                    */
/* As we only read/input the 1st 3 fields from the control file           */
/* and ignore the others, it will not become a data problem              */
/*****

/*****
/* S051: DATA FILENAME INCORRECT                                         */
/* Submission officer to manually check filename                           */
/* and inform the hospital of this error                                   */
/*****

/*****
/* S052: RECORD DATES AFTER CENSUS DATE - MODIFIED 1 JULY 2001          */
/*****

  if ((cdate < bookdate) or (cdate < sr_date) or (cdate < urg_date)
    or (cdate < reg_date) or (cdate < rem_date) or (cdate < urgup_dt)) then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S052 Record Dates After Census Date';
  end;

/*****
/* S053: SEE PREVIOUS MONTH COMPARISION EDITS                            */
/*****

/*****
/* S060: BLANK UNIQUE KEY                                                 */
/*****

  if key = '' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S060 Blank Unique Key';
  end;

```

```

/*****/
/* S065: BLANK PATIENT IDENTIFIER */
/*****/

if urno = '' then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S065 Blank Patient Identifier';
end;

/*****/
/* S066: INVALID PATIENT IDENTIFIER */
/*****/

tot=0;

do a = 1 to length(urno);
  tot =
indexc(substr(urno,a,1), '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrs
tuvwxyz ') + tot;
end;

if (tot/length(urno) ne 1) or (length(urno) > 10) then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S066 Invalid Patient Identifier';
end;

/*****/
/* S080: BLANK MEDICARE NUMBER */
/*****/

if (rem_date ne .) and (rmreason eq 'W') and
((medsuf = '') or (medsuf not in ('C-U', 'N-E', 'P-N', 'BAB'))) and (mednum =
'') then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S080 Blank Medicare Number';
end;

/*****/
/* S081: INVALID MEDICARE NUMBER */
/*****/

tot = 0;

do a = 1 to 11;
  tot = indexc(substr(mednum,a,1), ' 0123456789') + tot;
end;

if tot ne 11 and rem_date ne . and rmreason eq 'W' then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S081 Invalid Medicare Number';
end;

```

```

/*****
/* S082: INVALID COMBINATION MEDICARE CODE & DOB */
/*****

    if (rem_date ne .) and (rem_date >= '01Aug2000'd) and (substr(mednum,11,1) =
'0')
        and (intck('month', birthdat, cdate) > 6) then do;
            type3 = type3 + 1;
            type=type + 1;
            mess{type} = 'Type3:S082 Invalid Combination Medicare Code & DOB';
        end;

/*****
/* S083: INVALID COMBINATION MEDICARE CODE & DOB */
/*****

    if (rem_date ne .) and (rem_date >= '01Aug2000'd) and
        intck('day', birthdat, cdate) >= 365 and
        substr(mednum,11,1) = '0' then do;
            type1 = type1 + 1;
            type=type + 1;
            mess{type} = 'Type1:S083 Invalid Combination Medicare Code & DOB';
        end;

/*****
/* S087: BLANK MEDICARE SUFFIX */
/*****

    if reg_date = > '01Jul2000'd and (rem_date ne .) and
        (rmreason eq 'W') and (medsuf = '') then do;
            type1 = type1 + 1;
            type=type + 1;
            mess{type} = 'Type1:S087 Blank Medicare Suffix';
        end;

/*****
/* S088: INVALID MEDICARE SUFFIX */
/*****

    if reg_date = > '01Jul2000'd and (rem_date ne .) and rmreason eq 'W'
        and medsuf ne ''
        and (substr(medsuf,1,1) not in
            ('A','B','C','D','E','F','G','H','I','J',
            'K','L','M','N','O','P','Q','R','S','T',
            'U','V','W','X','Y','Z'))
        or (substr(medsuf,2,1) not in
            ('A','B','C','D','E','F','G','H','I','J',
            'K','L','M','N','O','P','Q','R','S','T',
            'U','V','W','X','Y','Z','-',' ',''))
        or (substr(medsuf,3,1) not in
            ('A','B','C','D','E','F','G','H','I','J',
            'K','L','M','N','O','P','Q','R','S','T',
            'U','V','W','X','Y','Z',' ','-')) then do;
            type1 = type1 + 1;
            type=type + 1;
            mess{type} = 'Type1:S088 Invalid Medicare Suffix';
        end;

```

```

/*****
/* S090: BLANK SEX CODE */
/*****

  if sex = '' then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S090 Blank Sex Code';
  end;

/*****
/* S091: INVALID SEX CODE */
/*****

  if sex not in ('1','2','3') then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S091 Invalid Sex Code';
  end;

/*****
/* S092: INDETERMINATE SEX, AGE >= 90 DAYS */
/*****

  if sex = '3' and intck('day',birthdat,reg_date) >= 90 then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S092 Indeterminate Sex, Age >= 90 Days';
  end;

/*****
/* S093: INDETERMINATE SEX - CHECK */
/*****

  if sex = '3' and intck('day',birthdat,reg_date) < 90 then do;
    type3 = type3 + 1;
    type=type + 1;
    mess{type} = 'Type3:S093 Indeterminate Sex - Check';
  end;

/*****
/* S095: BLANK DATE OF BIRTH */
/*****

  if birthdat = . then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S095 Blank Date of Birth';
  end;

/*****
/* S096: INVALID DATE OF BIRTH */
/*****

  dobcc = substr(put(birthdat, ddmmyy10.),7,2);
  if birthdat ne . and dobcc not in ('18','19','20') then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S096 Invalid Date of Birth';
  end;

```

```

/*****
/* S098: AGE GREATER THAN 120 YEARS */
/*****

if intck('year',birthdat,reg_date) >= 120 then do;
  type2 = type2 + 1;
  type=type + 1;
  mess{type} = 'Type2:S098 Age Greater Than 120 Years';
end;

/*****
/* S099: REGISTRATION DATE BEFORE DATE OF BIRTH */
/*****

if reg_date ne . and reg_date < birthdat then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S099 Registration Date Before Date of Birth';
end;

/*****
/* S110: BLANK WAITING NUMBER */
/*****

if waitno = '' then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S110 Blank Waiting Number';
end;

/*****
/* S111: INVALID WAITING NUMBER */
/*****

if waitno ne '' and
  (substr(waitno,1,1) not in ('0','1','2','3','4','5','6','7','8','9') or
  substr(waitno,2,1) not in ('0','1','2','3','4','5','6','7','8','9',' '))
then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S111 Invalid Waiting Number';
end;

if waitno ne '' and (waitno = '00' or waitno = '0') then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S111 Invalid Waiting Number';
end;

/***** SLA assignment *****/

format
sla_1 sla_2 sla_3 sla_4 sla_5 $5.
pw_1 pw_2 pw_3 pw_4 pw_5 8.
psloc_ok loc_ok pst_ok $1.
pst_loc $34.;

```

```

/*****
/* S115: AREA OF RESIDENCE ERROR */
/*****

  if reg_date = > '01Jul2000'd and (((locality = '') and (pstcode ne 1000 and
    pstcode ne 8888 and pstcode ne 9988))or ((locality ne '') and
    (pstcode = 1000 or pstcode = 9988))) then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S115 Area of Residence Error';
  end;

/*****
/* S116: INVALID LOCALITY (part 1) */
/*****

  if reg_date = > '01Jul2000'd and (locality) ne '' and pstcode=':3' then do;
    h_label=put(locality,loc00fma.);
    loc_ok = scan(h_label,1);
    if loc_ok ne '1' then do;
      type2 = type2 + 1;
      type=type + 1;
      mess{type} = 'Type2:S116 Invalid Locality';
    end;
  end;

/*****
/* S116: INVALID LOCALITY (part 2) */
/*****

  if reg_date = > '01Jul2000'd and verify(upcase(locality),
    "1234567890QWERTYUIOPASDFGHJKLZXCVBNM".- ") ne 0 then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S116 Invalid Locality';
  end;

/*****
/* S117: BLANK LOCALITY (part 2) */
/*****

  if reg_date = > '01Jul2000'd and (((locality = '') and (pstcode ne 1000 and
    pstcode ne 8888 and pstcode ne 9988))or ((locality ne '') and
    (pstcode = 1000 or pstcode = 9988))) then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S115 Area of Residence Error';
  end;

/*****
/* S120: BLANK POSTCODE */
/*****

  if pstcode = '' then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S120 Blank Postcode';
  end;

```

```

/*****
/* S121: INVALID POSTCODE
/*****

p_label=put(pstcode,pc00fma.);
pst_ok = scan(p_label,1);
if reg_date = > '01Jul2000'd and pst_ok ne '1' then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S121 Invalid Postcode';
end;

if pst_ok = '1' AND loc_ok = '1' then do;
    pst_loc=trim(pstcode)||trim(locality);
    g_label=put(pst_loc,sla00fma.);
    psloc_ok= scan(g_label,1,',');
    sla_1= scan(g_label,2,','); pw_1= scan(g_label,3,',');
    sla_2= scan(g_label,4,','); pw_2= scan(g_label,5,',');
    sla_3= scan(g_label,6,','); pw_3= scan(g_label,6,',');
    sla_4= scan(g_label,8,','); pw_4= scan(g_label,9,',');
    sla_5= scan(g_label,10,','); pw_5= scan(g_label,11);
end;

/*****
/* S122: INVALID COMBINATION POSTCODE AND LOCALITY
/*****

if reg_date = > '01Jul2000'd and psloc_ok ne '1' and pstcode=:'3' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S122 Invalid Combination Postcode And Locality';
end;

/***** SLA ASSIGNMENT WILL GO HERE *****/

if psloc_ok = '1' then do;
    t1 = pw_1;
    t2 = t1 + pw_2;
    t3 = t2 + pw_3;
    t4 = t3 + pw_4;
    t5 = t4 + pw_5;

    pw = 100*ranuni(4);

    if 0 <= pw <= t1 then sla98 = sla_1;
    else if t1 < pw <= t2 then sla98 = sla_2;
    else if t2 < pw <= t3 then sla98 = sla_3;
    else if t3 < pw <= t4 then sla98 = sla_4;
    else if t4 < pw <= 100 then sla98 = sla_5;
    else sla98='error';

end;

/***** end sla assignment *****/

```

```

/*****
/* S133: BLANK PRINCIPAL PRESCRIBED PROCEDURE */
/*****

if ppproc = '' then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S133 Blank Principal Prescribed Procedure';
end;

/*****
/* S134: INVALID PRINCIPAL PRESCRIBED PROCEDURE */
/*****
if reg_date ne . and reg_date < '01Jul2000'd and ppproc ne '' then do;
  if ppproc ne '' and ppproc not in
    ('01', '02', '03', '04', '05', '06', '07', '08', '10', '11',
     '12', '13', '14', '15', '16', '17', '18', '19', '20', '21',
     '22', '30', '32', '33', '34', '35', '36', '37', '38', '39',
     '40', '41', '42', '50', '52', '54', '55', '56', '57', '58',
     '59', '60', '61', '62', '70', '71', '72', '73', '74', '75',
     '77', '79', '81', '82', '83', '84', '85', '86', '90', '91',
     '92', '93', '94', '95', '96', '97', '98', '99',
     '100', '110', '111', '112', '113', '114', '115', '116', '117',
     '118', '119', '120', '121', '130', '131', '132', '133', '134',
     '135', '136', '137', '138', '140', '141', '142', '143', '150',
     '160', '161', '162', '163', '164', '165', '166', '167', '168',
     '169', '170', '171', '172', '173', '174', '175', '176', '180',
     '181', '182', '183', '184', '185', '186', '187', '188', '190',
     '191', '192', '193', '194', '195', '196', '197', '198', '199',
     '202', '203', '204', '205', '206', '207', '208', '209', '210',
     '211', '212', '213', '214', '215', '216', '217', '218', '219',
     '220', '221', '222', '223', '224', '225', '226') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S134 Invalid Principal Prescribed Procedure';
  end;
end;

/*****
/* S135: PATIENT ALREADY ON WAITING LIST FOR SAME PPP */
/*****

if ppp_dup > 1 then do;
  type3 = type3 + 1;
  type=type + 1;
  mess{type} = 'Type3:S135 Patient Already On Waiting List For Same PPP';
end;

/*****
/* S136: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S146: BLANK SURGICAL SPECIALTY */
/*****

if surgspec = '' then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S146 Blank Surgical Specialty';
end;

```

```

/*****
/* S147: INVALID SURGICAL SPECIALTY */
/*****

    if surgspec not in ('01','02','03','04','05','06','07','08',
        '09','10','11') then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S147 Invalid Surgical Specialty';
    end;

/*****
/* S148: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S160: BLANK CLINICAL URGENCY CATEGORY */
/*****

    if urgency = '' then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S160 Blank Clinical Urgency Category';
    end;

/*****
/* S161: INVALID CLINICAL URGENCY CATEGORY */
/*****

    if urgency ne '' and urgency not in ('1','2','3') then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S161 Invalid Clinical Urgency Category';
    end;

/*****
/* S162: CLINICAL URGENCY REASSIGNMENT INCOMPLETE */
/*****

    if (old_urg ne '' and urg_date = .) or
        (old_urg = '' and urg_date ne .) then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S162 Clinical Urgency Reassignment Incomplete';
    end;

/*****
/* S166: BLANK PLANNED length OF STAY */
/*****

    if planstay = '' then do;
        type2 = type2 + 1;
        type=type + 1;
        mess{type} = 'Type2:S166 Blank Planned length of Stay';
    end;

```

```

/*****
/* S167: INVALID PLANNED length OF STAY */
/*****

  if planstay ne '' and planstay not in ('1','2') then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S167 Invalid Planned length of Stay';
  end;

/*****
/* S168: BLANK REGISTRATION DATE */
/*****

  if reg_date = . then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S168 Blank Registration Date';
  end;

/*****
/* S169: INVALID REGISTRATION DATE */
/*****

  regcc = substr(put(reg_date, ddmmyy10.),7,2);
  if reg_date ne . and regcc not in ('19','20') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S169 Invalid Registration Date';
  end;

/*****
/* S170: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S171: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S172: REG DATE = URGENCY REASSIGNMENT DATE OR
      DATE OF LAST CLINICAL URGENCY INCREASE - ADDED 1 JULY 2001 */
/*****

  if reg_date = > '01Jul1999'd and ((reg_date = urg_date) or
    (reg_date = urgup_dt)) then do;
    type3 = type3 + 1;
    type=type + 1;
    mess{type} = 'Type3:S172 Reg Date = Urgency Reassignment Date/Urgency
Increase Date';
  end;

/*****
/* S173: DATE OF REG = STATUS REASSIGNMENT DATE */
/*****

  if reg_date = > '01Jul2000'd and reg_date = sr_date then do;
    type3 = type3 + 1;
    type=type + 1;
    mess{type} = 'Type3:S173 Reg Date = Status Reassignment Date';
  end;

```

```

/*****
/* S192: BLANK SOURCE OF REFERRAL */
/*****

if reg_date = > '01Jul2000'd and refsourc = '' then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S192 Blank Source of Referral';
end;

/*****
/* S193: INVALID SOURCE OF REFERRAL */
/*****

if reg_date = > '01Jul2000'd and (refsourc) ne '' and (refsourc) not in
('1','2','3','4','5') then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S193 Invalid Source of Referral';
end;

/*****
/* S194: BLANK REFERRING HOSPITAL */
/*****

if reg_date = > '01Jul2000'd and refsourc = '2' and refhosp = '' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S194 Blank Referring Hospital';
end;

/*****
/* S195: INVALID REFERRING HOSPITAL */
/*****

if reg_date = > '01Jul1999'd and (refsourc = '2' and refhosp ne ''
and refhosp not in
('1010','1590','1031','2010','2050','1020','1050','2110','2111',
'2220','1121','1210','1170','2440','1280','1191','1331',
'1240','1230','1360','1450','1390','1150','1580','1180') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S195 Invalid Referring Hospital';
end;

/*****
/* S196: REFERRING HOSPITAL CONFLICT */
/*****

if reg_date = > '01Jul2000'd and refsourc not in ('2') and refhosp ne '' then
do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S196 Referring Hospital Conflict';
end;

```

```

/*****
/* S199: INVALID DATE OF LAST CLINICAL URGENCY INCREASE */
/* ADDED 1 JULY 2001 */
/*****

urgcc = substr(put(urgup_dt, ddmmyy10.),7,2);
if urgup_dt ne . and urgcc not in ('19','20') then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S199 Invalid Date of Last Clinical Urgency Increase';
end;

/*****
/* S260: INCOMPLETE URGENCY REASSIGNMENT - DELETED 1 JULY 2000 */
/*****

/*****
/* S261: PREV URGENCY CATEGORY = CLINICAL URGENCY */
/*****

if old_urg ne ' ' and urgency ne ' ' and old_urg = urgency then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S261 Prev Urgency Category = Clinical Urgency';
end;

/*****
/* S262: INVALID PREVIOUS URGENCY CATEGORY */
/*****

if urg_date ne . and old_urg ne ' ' and
  old_urg not in ('1','2','3') then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S262 Invalid Previous Urgency Category';
end;

/*****
/* S263: INVALID URGENCY REASSIGNMENT DATE */
/*****

urgcc = substr(put(urg_date, ddmmyy10.),7,2);
if urg_date ne . and urgcc not in ('19','20') then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S263 Invalid Urgency Reassignment Date';
end;

/*****
/* S264: BLANK PATIENT LISTING STATUS */
/*****

if pat_stat = ' ' then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S264 Blank Patient Listing Status';
end;

```

```

/*****
/* S265: INVALID PATIENT LISTING STATUS */
/*****

  if pat_stat ne '' and pat_stat not in ('N','R') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S265 Invalid Patient Listing Status';
  end;

/*****
/* S266: INVALID STATUS REASSIGNMENT DATE */
/*****

  srcc = substr(put(sr_date, ddmmyy10.),7,2);
  if sr_date ne . and srcc not in ('19','20') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S266 Invalid Status Reassignment Date';
  end;

/*****
/* S267: INVALID REASON FOR NRFC STATUS */
/*****

  if pat_stat = 'N' and nrreason ne '' and nrreason not in ('C','D','S') then
do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S267 Invalid Reason For NRFC status';
  end;

/*****
/* S268: PATIENT NRFC, NO REASON FOR NRFC */
/*****

  if pat_stat = 'N' and nrreason = '' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S268 Patient NRFC, No Reason For NRFC';
  end;

/*****
/* S269: REASON FOR NRFC PATIENT CURRENT RFC */
/*****

  if pat_stat = 'R' and nrreason ne '' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S269 Reason For NRFC Patient Current RFC';
  end;

/*****
/* S274: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

```

```

/*****
/* S275 INVALID BOOKING DATE */
/*****

bookcc = substr(put(bookdate, ddmmyy10.),7,2);
if bookdate ne . and bookcc not in ('19','20') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S275 Invalid Booking Date';
end;

/*****
/* S276: BOOKING DATE BEFORE REGISTRATION DATE */
/*****

if reg_date = > '01Jul1999'd and bookdate ne . and bookdate < reg_date then
do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S276 Booking Date Before Registration Date';
end;

/*****
/* S277: INVALID BOOKING NUMBER */
/*****

if bookno ne . and bookno not in
('00','01','02','03','04','05','06','07','08','09','10',
'11','12','13','14','15','16','17','18','19','20') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S277 Invalid Booking Number';
end;

/*****
/* S278: INCOMPLETE BOOKING DETAILS */
/*****

if sch_date ne '' then do;
    if (bookdate eq '') or (bookno eq '') then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S278 Incomplete Booking Details';
    end;
end;

/*****
/* S279: INVALID REASON FOR REBOOKING */
/*****

if bookno > 1 and bookno ne '' and
rbreason not in ('A','X','C','D','H','P','F') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S279 Invalid Reason For Rebooking';
end;

if bookno = 1 and sch_date eq '' and
rbreason not in ('A','X','C','D','H','P','F') then do;
    type1 = type1 + 1;
    type=type + 1;

```

```
    mess{type} = 'Type1:S279 Invalid Reason For Rebooking';  
end;
```

```

/*****
/* S280: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S281: BOOKING DATE PRESENT/ BOOKING NUMBER ZERO */
/* DELETED 1 JULY 2001 */
/*****

/*****
/* S282: SCHEDULED ADMISSION DATE PRESENT/BOOKING NUMBER ZERO */
/* DELETED 1 JULY 2001 */
/*****

/*****
/* S283: BOOKING/SCHEDULED ADMISSION DATE AND BOOKING NUMBER MISMATCH */
/*****

    if ((sch_date ne . or bookdate ne .) and bookno in ('00','0')) or
        (bookno gt 0 and bookdate eq .) then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S283 Booking/Scheduled Admission Date and Booking Number
mismatch';
    end;

/*****
/* S285: INVALID SCHEDULED ADMISSION DATE */
/*****

    schcc = substr(put(sch_date, ddmmyy10.),7,2);
    if schcc not in ('19','20') and sch_date ne . then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S285 Invalid Scheduled Admission Date';
    end;

/*****
/* S286: SCHEDULED ADM DATE BEFORE BOOKING DATE */
/*****

    if sch_date < bookdate and sch_date ne . then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S286 Scheduled Adm Date Before Booking Date';
    end;

/*****
/* S287: SCHEDULED ADMISSION DATE EXCEEDDED */
/*****

    if rmreason not in ('E','X','W') and sch_date ne .
        and sch_date <= rcddate and rem_date = . then do;
        type1 = type1 + 1;
        type=type + 1;
        mess{type} = 'Type1:S287 Scheduled Admission Date Exceeded';
    end;

```

```

/*****
/* S290: INVALID REMOVAL DATE */
/*****

remcc = substr(put(rem_date, ddmmyy10.),7,2);
if remcc not in ('19','20') and rem_date ne . then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S290 Invalid Removal Date';
end;

/*****
/* S291: REMOVAL DATE PRIOR TO REG DATE */
/*****

if rem_date ne . and rem_date < reg_date then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S291 Removal Date Prior To Reg Date';
end;

/*****
/* S292: REMOVED BUT NO REASON */
/*****

if rem_date ne . and rmreason = '' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S292 Removed But No Reason';
end;

/*****
/* S293: REMOVED, NO REMOVAL DATE */
/*****

if rmreason in ('X','W','M','B','I','U','R','Z','Q','F','O','N','T')
and rem_date = . then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S293 Removed, No Removal Date';
end;

/*****
/* S294: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S295: REMOVAL DATE NOT = SCHEDULED ADMISSION DATE */
/*****

if rmreason in ('X','W') and rem_date ne .
and rem_date ne sch_date then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S295 Removal Date Not = Scheduled Admission Date';
end;

```

```

/*****
/* S296: PATIENT ADMITTED, NOT READY FOR CARE */
/*****

if rmreason in ('X','W') and rem_date ne . and pat_stat = 'N' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S296 Patient Admitted, Not Ready For Care';
end;

/*****
/* S298: INVALID REASON FOR REMOVAL */
/*****

S298 = 'N';

if rmreason not in ('X','W','M','B','I','U','R','Z','Q','F','O',
    'N','T','') then do;
    S298 = 'Y';
end;

if (rmreason not in ('X','W') and S298 eq 'N'
and rem_date > '01Jul2001'd) then do;
    if rem_date ne . and rmreason ne '' and
        rmreason not in
            ('X','W','M','B','I','U','R','Z','Q','F','O','N','T') then do;
        S298 = 'Y';
    end;
end;

if S298 = 'Y' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S298 Invalid Reason For Removal';
end;

/*****
/* S299: Cancelled, no reason for cancellation - removed 1 July 2001 */
/* S300: Invalid comb: removal and cancellation - removed 1 July 2001 */
/* S301: Invalid reason for cancellation - removed 1 July 2001 */
/*****

/*****
/* S302: INVALID INSURANCE DECLARATION */
/*****

if insure ne '' and insure not in
    ('M','P','V','W','T','A','S','C','O','X') then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S302 Invalid Insurance Declaration';
end;

```

```

/*****/
/* S303: BLANK INSURANCE DECLARATION */
/*****/

if rmreason eq 'W' and insure = '' then do;
    type2 = type2 + 1;
    type=type + 1;
    mess{type} = 'Type2:S303 Blank Insurance Declaration';
end;

/*****/
/* S309: BLANK TRANSFER DESTINATION */
/*
/*     MODIFIED 1 JULY 2001
/*
/*     * reg_date comparison date from 2000 to 1999
/*
/*     * include 'N' in rmreason code */
/*****/

if reg_date = > '01Jul1999'd and rmreason in ('T','N','X')
    and transdes = '' then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S309 Blank Transfer Destination';
end;

/*****/
/* S310: TRANSFER DESTINATION NOT TRANSFERRED */
/*
/*     MODIFIED 1 JULY 2001
/*
/*     * include 'N' in rmreason code */
/*****/

if reg_date = > '01Jul2001'd and transdes ne ''
    and rmreason not in ('T','N','X') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S310 Transfer Destination Not Transferred';
end;

/*****/
/* S311: SEE PREVIOUS MONTH COMPARISION EDITS */
/* S315: SEE PREVIOUS MONTH COMPARISION EDITS */
/* S316: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****/

/*****/
/* S318: TOTAL NRFC DAYS SINCE LAST CLINICAL URGENCY INCREASE - */
/*     NOT NUMERIC OR BLANK */
/*
/*     ADDED 1 JULY 2001 */
/*****/

if not verify(nrtot_ui, '0123456789.') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S318 Total NRFC Days /Date of Last Clinical Urgency
Increase Not Numeric/Blank';
end;

```

```

/*****
/* S319: TOTAL NRFC LESS THAN TOTAL SINCE LAST URGENCY INCREASE */
/*****

  if nrtot_ui > nr_total then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type1:S319 Total NRFC less than total since last urgency
increase';
  end;

/*****
/* S320: TOTAL NRFC DAYS NOT NUMERIC OR BLANK */
/*****

  if not verify(nr_total, '0123456789.') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S320 Total NRFC Days Not Numeric Or Blank';
  end;

/*****
/* S321: TOT NRFC DAYS FOLLOWING LAST URG REASS NOT NUMERIC OR BLANK */
/*****

  if not verify(nr_assig, '0123456789.') then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S321 Tot NRFC Days Following Last Urg Reass Not
Numeric Or Blank';
  end;

/*****
/* S322: TOTAL NRFC LESS THAN TOTAL SINCE URG REASSIGN */
/*****

  if nr_assig > nr_total then do;
    type1 = type1 + 1;
    type=type + 1;
    mess{type} = 'Type1:S322 Total NRFC Less Than Total Since Urg Reassign';
  end;

/*****
/* S323: SEE PREVIOUS MONTH COMPARISION EDITS */
/* S324: SEE PREVIOUS MONTH COMPARISION EDITS */
/* S325: SEE PREVIOUS MONTH COMPARISION EDITS */
/* S326: REMOVED AND REPLACED WITH S336 S337 S338 1 JUNE 2001 (JB) */
/* S327: REMOVED AND REPLACED WITH S336 S337 S338 1 JUNE 2001 (JB) */
/*****

```

```

/*****
/* S328: BLANK HOSPITAL INITIATED POSTPONEMENT */
/*****

if hi_post = '' then do;
  type2 = type2 + 1;
  type=type + 1;
  mess{type} = 'Type2:S328 Blank Hospital Initiated Postponement';
end;

/*****
/* S329: INVALID COMBINATION REASON FOR REBOOKING
*/
/*          AND HOSPITAL INITIATED POSTPONEMENT */
/*****

if rmreason ne 'E' and rbreason in ('H','D') and bookdate ne .
  and (hi_post = '' or hi_post = '0') then do;
  type3 = type3 + 1;
  type=type + 1;
  mess{type} = 'Type3:S329 Invalid comb Reason for Rebooking & Hosp Initiated
Post';
end;

/*****
/* S330: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S331: BOOKING NUMBER < HOSPITAL INITIATED POSTPONEMENT */
/*****

if bookno ne . and bookno < hi_post then do;
  type1 = type1 + 1;
  type=type + 1;
  mess{type} = 'Type1:S331 Booking Number < Hospital Initiated Postponement';
end;

/*****
/* S332: SEE PREVIOUS MONTH COMPARISION EDITS */
/*****

/*****
/* S333: Invalid Hospital Initiated Postponement */
/*****

if bookdate ne . and bookdate < '01Jul2000'd
  and hi_post not in ('0','00') and rbreason not in ('H','D') then do;
  type3 = type3 + 1;
  type=type + 1;
  mess{type} = 'Type3:S333 Invalid Hospital Initiated Postponement';
end;

/*****
/* S334: Hosp Initiated Post Not Zero,
*/
/*          Rebook Reason not Hosp Initiated */
/*****

if rmreason ne 'E' and bookdate ne . and bookdate > '30Jun2000'd
  and rbreason not in ('H','D') and hi_post not in ('0','00') then do;

```

```
type3 = type3 + 1;
type=type + 1;
mess{type} = 'Type3:S334 Hosp Initiated Post Not Zero, Rebook Reason not
Hosp Initiated';
end;
```

```

/*****
/* S335: SEE PREVIOUS MONTH COMPARISION EDITS */
*****/

/***** ADD VARIABLES *****/
format waitdays 4. tot_wait 4.;
if pat_stat = 'N' then w_stat = 'N';

/*REMOVED FROM WAITING LIST*/
if rmreason ne '' and rem_date ne . then do;
  if rmreason in ('W','X') AND (rem_date ne .)
    then w_stat = 'A'; /* if they were admitted */
  else if rmreason in ('R','Z','Q','F','O','B','I','U','M')
    then w_stat = 'C'; /* if they were
cancelled/removed */
  else if rmreason in ('E')
    then w_stat = 'E'; /* if they were data error */
  else if rmreason in ('N','T')
    then w_stat = 'T'; /* if they were transferred */

/* 31/08/2001: After discussion with Tim Williamson, Mark Gill & Jenelle Hurley
rmreason 'M' is to be included in w_stat cancelled because it is an UNPLANNED
admission.
3/09/2001: Agreed to relabel the W_Stat value 'C' to rem_canc.
*/

/* do not have urgdate or urg increase date*/
if (urg_date=. and urgup_dt=.) then do;
  waitdays = intck('day', reg_date, rem_date );
  waityrs = intck('year', reg_date, rem_date );
end;

/*urgency/increase date changed before 1 July 2001 */
if urg_date <= '30JUN2001'd and urgup_dt = . and urg_date ne . then do;
  waitdays = intck('day', urg_date, rem_date);
  waityrs = intck('year', urg_date, rem_date);
end;

/*Urgency increase after 1 July 2001*/
if urgup_dt ne . then do;
  waitdays = intck('day',urgup_dt, rem_date);
  waityrs = intck('year',urgup_dt, rem_date);
end;

/*Urgency decrease after 1 July 2001 and June data contains an
urgency date (Uses June urgency date to calculate waitdays
and waityrs)*/
if urg_date >= '01JUL2001'd and ((urgup_dt eq .) and
(junurgdt ne .)) then do;
  waitdays = intck('day',junurgdt, rem_date);
  waityrs = intck('year',junurgdt, rem_date);
end; /*Urgency decrease after 1 July 2001 and June data does not
contain an urgency data (Uses registration date to calculate
waitdays and waityrs)*/
else if urg_date >= '01JUL2001'd and ((urgup_dt eq .) and
(junurgdt = .)) then do;
  waitdays = intck('day',reg_date, rem_date);
  waityrs = intck('year',reg_date, rem_date);
end;
end; /*end removed from waiting list*/

```

```

/*STILL ON WAITING LIST*/
if (rmreason = ' ' and rem_date = .) then do;
  w_stat = 'W';
  if pat_stat = 'N' then w_stat = 'N';

  /* do not have urgdate/urgency increase date */
  if (urg_date=. and urgup_dt=.) then do;
    waitdays = intck('day', reg_date, cdate);
    waityrs = intck('year', reg_date, cdate);
  end;

  /*Urgency/increase change before 1 July 2001*/
  if urg_date <= '30JUN2001'd and urgup_dt = . and urg_date ne . then do;
    waitdays = intck('day', urg_date, cdate);
    waityrs = intck('year',urg_date, cdate);
  end;

  /*Urgency increase after 1 July 2001*/
  if urgup_dt ne . then do;
    waitdays = intck('day',urgup_dt, cdate);
    waityrs = intck('year',urgup_dt, cdate);
  end;

  /*Urgency decrease after 1 July 2001*/
  if urg_date >= '01JUL2001'd and ((urgup_dt = .) and
    (junurgdt ne .)) then do;
    waitdays = intck('day', junurgdt, cdate);
    waityrs = intck('year', junurgdt, cdate);
  end;
  else if urg_date >= '01JUL2001'd and ((urgup_dt = .) and
    (junurgdt = .)) then do;
    waitdays = intck('day',reg_date, cdate);
    waityrs = intck('year',reg_date, cdate);
  end;
end; /*end still on waiting list*/

/** Calculate Total Waiting Time for those who do not have urgdate/urg increase
date ***/

if urg_date = . and urgup_dt =. and reg_date ne . then do;
  if w_stat in ('N','W') then tot_wait = (waitdays - nr_total) + 1;
  if w_stat in ('C','E','T','A') then tot_wait = (waitdays - nr_total);
end;

/**/ Calculate Total Waiting Time for those who have urgdate before 1 July 2001
***/

if urg_date <= '30JUN2001'd and urgup_dt = . and reg_date ne . and urg_date ne
. then do;
  if w_stat in ('N','W') then tot_wait = (waitdays - /*nr_assig*/nrassold) +
1;
  if w_stat in ('C','E','T','A') Then tot_wait = (waitdays -
/*nr_assig*/nrassold);
end;

```

```

/** Calculate Total Waiting Time for those who have urgency increase after 1
July 2001 */

if urgup_dt ne . and reg_date ne . then do;
  if w_stat in ('N','W') then tot_wait = (waitdays - nrtot_ui) + 1;
  if w_stat in ('C','E','T','A') then tot_wait = (waitdays - nrtot_ui);
end;

/** Calculate Total Waiting Time for those who have urgency decrease after 1
July 2001 */

if urg_date >= '01JUL2001'd and ((urgup_dt = .) /*and urg_date ne .*/)
and reg_date ne . and reg_date < '01JUL2001'd then do;
  if w_stat in ('N','W') then tot_wait = (waitdays - junnrass) + 1;
  if w_stat in ('C','E','T','A') then tot_wait = (waitdays - junnrass);
end;

/** Calculate Total Waiting Time for those who have urgency decrease after 1
July 2001
and registration date is after 01 July 2001
*/

if nr_total eq . then nr_total = 0;

if urg_date >= '01JUL2001'd and ((urgup_dt = .) and urg_date ne .)
and reg_date >= '01JUL2001'd then do;
  if w_stat in ('N','W') then tot_wait = (waitdays - nr_total) + 1;
  if w_stat in ('C','E','T','A') then tot_wait = (waitdays - nr_total);
end;

/***** END TOTAL WAITING TIME CALCULATIONS *****/

OVERADMT = 'N';
OVERWAIT = 'N';
WAITSTAT = 'N';

/* Patients remaining on the WAITING LIST have waited more than the desirable
time */

if w_stat in ('W') then do;
  if urgency = 1 then do;
    if tot_wait > 30 then OVERWAIT = 'Y';
  end;
  else if urgency = 2 then do;
    if tot_wait > 90 then OVERWAIT = 'Y';
  end;
end;

/* Patients who have waited more than the desirable time PRIOR TO ADMISSION */

if w_stat = 'A' and rem_date ne . and rmreason in ('X','W') then do;
  if urgency = 1 then do;
    if tot_wait > 30 then OVERADMT = 'Y';
  end;
  else if urgency = 2 then do;
    if tot_wait > 90 then OVERADMT = 'Y';
  end;
end;

```

```

/***** Assign waitstat *****/

if w_stat in ('N') AND
  rmreason in ('X','W','M','B','I','U','R','Z','Q','F','O','N','T')
then waitstat = 'N';
else if w_stat in ('W','N') then waitstat = 'Y';
  else if w_stat in ('C','A') then waitstat = 'N';
    else waitstat = 'E';

/*****Extra edits*****/

/*****
/* S311: WAIT EQUALS OVER FIVE YEARS */
*****/

if waityrs >= 5 then do;
  type3 = type3 + 1;
  type=type + 1;
  mess{type} = 'Type3:S311 Wait Equals Five Years or more';
end;

/*****
/* S315: CLINICAL URG CAT 1, WAITING > 30 DAYS */
*****/

if urgency = '1' and (overadmt = 'Y' or overwait = 'Y') then do;
  type3 = type3 + 1;
  type=type + 1;
  mess{type} = 'Type3:S315 Clinical Urg Cat 1, Waiting > 30 Days';
end;

/*****
/* S316: CLINICAL URG CAT 2, WAITING > 90 DAYS */
*****/

if urgency = '2' and (overadmt = 'Y' or overwait = 'Y') then do;
  type3 = type3 + 1;
  type=type + 1;
  mess{type} = 'Type3:S316 Clinical Urg Cat 2, Waiting > 90 Days';
end;

/*****
/* S323: TOTAL NRFC DAYS NOT EQUAL TO ZERO */
*****/

if reg_date = sr_date and pat_stat = 'R'
  and nr_total ne 0 then do;
  type1 = type1 + 1;
  type = type + 1;
  mess{type} = 'Type1:S323 Total NRFC Days not Equal to Zero';
end;

/*****
/* S324: NEGATIVE WAITING TIME (NO URGENCY REASSIGNMENT) */
*****/

if urg_date = . and tot_wait < 0 then do;
  type1 = type1 + 1;
  type = type + 1;
  mess{type} = 'Type1:S324 Negative Waiting Time (No Urgency Reassignment)';
end;

```

```

/*****/
/* S325: NEGATIVE WAITING TIME (URGENCY REASSIGNMENT) */
/*****/

if urg_date ne . and tot_wait < 0 then do;
  type1 = type1 + 1;
  type = type + 1;
  mess{type} = 'Type1:S325 Negative Waiting Time (Urgency Reassignment)';
end;
/*****/
/* S327: REMOVED 1 JULY 2001 */
/*****/

/*****/
/* S345: DATE OF LAST CLINICAL URGENCY INCREASE < 1 JULY 2001
*/
/*      ADDED 1 JULY 2001 */
/*****/
if urgup_dt ne . and urgup_dt <= '30JUN2001'd then do;
  type1 = type1 + 1;
  type = type + 1;
  mess {type} = 'Type1:S345 Date of Last Clinical Urgency Increase < 1 July
2001';
end;

/*****/
/* S346: URGENCY REASSIGNMENT DATE < DATE OF LAST CLINICAL - */
/*      URGENCY INCREASE */
/*      ADDED 1 JULY 2001 */
/*****/
if (urg_date < urgup_dt) or (urg_date eq . and urgup_dt ne .) then do;
  type1 = type1 + 1;
  type = type + 1;
  mess {type} = 'Type1:S346 Urgency reass date < date of last clinical urgency
increase';
end;

if type1 = . then type1 = 0;
if type2 = . then type2 = 0;
if type3 = . then type3 = 0;
if type = . then type = 0;

x = intnx('month', cdate ,0); /*** FIRST DAY OF CENSUS MONTH ***/

end; /* END CHECKS AGAINST NEW DATA ONLY */
/***** COMPARISONS *****/

/*****/
/* S136: PROCEDURE MISMATCH BETWEEN MONTHS */
/*****/

if (sex = sexold) and (birthdat = birthold) and (ppproc ne procold) then do;
  type3 = type3 + 1;
  type = type + 1;
  mess{type} = 'Type3:S136 Procedure Mismatch Between Months';
end;

```

```

/*****
/* S148: SURGICAL SPECIALTY MISMATCH BETWEEN MONTHS */
/*****

  if (sex = sexold) and (birthdat = birthold) and (surgspec ne specold) then do;
    type3 = type3 + 1;
    type = type + 1;
    mess{type} = 'Type3:S148 Surgical Specialty Mismatch Between Months';
  end;

/*****
/* S170: DIFFERENT REGISTRATION DATE */
/*****

  if (sex = sexold) and (birthdat = birthold) and (reg_date ne regold) then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type1:S170 Different Registration Date';
  end;

/*****
/* S171: NEW RECORD, REG DATE PRIOR TO THIS MONTH */
/*****

  if reg_date ne . and (sexold = ' ') and (birthold = .) then do;
    if (reg_date < x) then do;          /*** x is first day of census month ***/
      type3 = type3 + 1;
      type = type + 1;
      mess{type} = 'Type3:S171 New record, Reg Date Prior To This Month';
    end;
  end;

/*****
/* S294: RECORD REMOVED LAST MONTH */
/*****

  if (wstatold = 'N') AND (sex = sexold) AND (birthdat = birthold) then do;
    type3 = type3 + 1;
    type = type + 1;
    mess{type} = 'Type3:S294 Record Removed Last Month';
  end;

/*****
/* S053: RECORD NOT REMOVED LAST MONTH MISSING FROM FILE */
/*****

  if (wstatold = 'Y') AND (sex = ' ') AND (birthdat = .) then do;
    type2 = type2 + 1;
    type = type + 1;
    mess{type} = 'Type2:S053 Record not removed last month missing from file';
  end;

```

```

/*****
/* S280: BOOKING NUMBER MISMATCH BETWEEN MONTHS */
/*****

if (wstatold = 'Y') AND (sex = sexold) AND (birthdat = birthold)
    AND (bookno < booknold) then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type1:S280 Booking Number mismatch between months';
end;

/*****
/* S330: HOSPITAL INITIATED POSTPONEMENT MISMATCH BETWEEN MONTHS */
/*****

if (wstatold = 'Y') AND (sex = sexold) AND (birthdat = birthold)
    AND (hi_post < hiposold) then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type1:S330 Hospital Initiated Postponement Mismatch Between
Months';
end;

/*****
/* S332: BOOKING NUMBER INCREASED, HOSPITAL INITIATED */
/* POSTPONEMENT UNCHANGED */
/*****

if (wstatold = 'Y') AND (sex = sexold) AND (birthdat = birthold)
    AND rbreason in ('H', 'D') AND bookno > booknold AND hi_post =
hiposold AND
    schold ne '' then do;
    type3= type3 + 1;
    type = type + 1;
    mess{type} = 'Type3:S332 Booking Number Increased, Hosp Initiated Post
Unchanged';
end;

/*****
/* S335: Hosp Initiated Post Increased, Rebook Reason not HIP */
/* This is from July 2000 data only */
/*****

if (wstatold = 'Y') AND (sex = sexold) AND (birthdat = birthold)
    AND rbreason not in ('H', 'D') AND hi_post > hiposold then do;
    type1 = type1 + 1;
    type = type + 1;
    mess{type} = 'Type3:S335 Hosp Initiated Post Increased, Rebook Reason not
Hosp Initiated';
end;

```

```

/*****
/* S336: TOTAL NOT READY FOR CARE DAYS MISMATCH                               */
/*      ADDED 1 JULY 2001                                                       */
/*****
rejs336 = 'N';

if pat_stat = 'R' and sr_date > reg_date and nr_total = 0
  then rejs336 = 'Y';

if pat_stat = 'N' and ((rem_date ne .) or
  (rmreason in ('X','W','M','B','I','U','R','Z','Q','F','O','N','T')))
  then do;
  if sr_date < x and (nr_total < (nrtotold + intck('day',x,rem_date)))
    then rejs336 = 'Y';
  if sr_date >= x and (nr_total < (nrtotold +
intck('day',sr_date,rem_date)))
    then rejs336 = 'Y';
  end;

if pat_stat = 'N' and rem_date eq . and rmreason eq ' ' and sr_date < x then
do;
  if month(cdate) in (7,8,10,12,1,3,5) then do;
    if not nr_total = nrtotold + 31 then rejs336 = 'Y';
  end;
  else if month(cdate) eq 2 then do;
    if not nr_total = nrtotold + 28 then rejs336 = 'Y';
  end;
  else if month(cdate) in (9,11,4,6) then do;
    if not nr_total = nrtotold + 30 then rejs336 = 'Y';
  end;
end;

if pat_stat = 'N' and rem_date eq . and rmreason eq ' ' and sr_date >= x then
do;
  if nr_total < (nrtotold + intck('day',sr_date,cdate) + 1)
    then rejs336 = 'Y';
  end;

if rejs336 eq 'Y' then do;
  typel = typel + 1;
  type = type + 1;
  mess {type} = 'Typel:S336 Total NRFC days mismatch';
end;

/*****
/* S337: TOTAL NRFC DAYS FOLLOWING LAST CLINICAL URGENCY                       */
/*      REASSIGNMENT MISMATCH                                                 */
/*      ADDED 1 JULY 2001                                                       */
/*****
if cdate > '31JUL2001'd then do;
  rejs337 = 'N';

if sr_date ne . and urg_date ne . then do;
  if pat_stat = 'R' and sr_date > urg_date and sr_date > reg_date and
    urg_date > reg_date and nr_assig = 0 then rejs337 = 'Y';

```

```

/***** S337 REMOVED FROM WAITING LIST *****/

if pat_stat = 'N' and ((rem_date ne .) or (rmreason in
('X','W','M','B','I',
 'U','R','Z','Q','F','O','N','T')) then do;
if urg_date >= x and sr_date >= x and urg_date < sr_date
and (nr_assig < (nrassold + intck('day',sr_date,rem_date)))
then rejs337 = 'Y';
if urg_date >= x and sr_date >= x and urg_date > sr_date
and (nr_assig < (nrassold + intck('day',urg_date,rem_date)))
then rejs337 = 'Y';
if urg_date < x and sr_date < x and
(nr_assig < (nrassold + intck('day',x,rem_date)))
then rejs337 = 'Y';
if urg_date >= x and sr_date < x and
(nr_assig < (nrassold + intck('day',urg_date,rem_date)))
then rejs337 = 'Y';
if urg_date < x and sr_date >= x and
(nr_assig < (nrassold + intck('day',sr_date,rem_date)))
then rejs337 = 'Y';
end;

/***** S337 REMAIN ON WAITING LIST *****/

if pat_stat = 'R' and sr_date > urg_date and sr_date > reg_date and
urg_date > reg_date and nr_assig = 0 then rejs337 = 'Y';

if pat_stat = 'N' and (rem_date eq . and rmreason eq '') then do;
if urg_date >= x and sr_date >= x and urg_date < sr_date
and (nr_assig < (nrassold + intck('day',sr_date,cdate)+1))
then rejs337 = 'Y';
if urg_date >= x and sr_date >= x and urg_date > sr_date
and (nr_assig < (nrassold + intck('day',urg_date,cdate)+1))
then rejs337 = 'Y';
if urg_date < x and sr_date < x and
(nr_assig < (nrassold + intck('day',x,cdate)+1))
then rejs337 = 'Y';
if urg_date >= x and sr_date < x and
(nr_assig < (nrassold + intck('day',urg_date,cdate)+1))
then rejs337 = 'Y';
if urg_date < x and sr_date >= x and
(nr_assig < (nrassold + intck('day',sr_date,cdate)+1))
then rejs337 = 'Y';
end;
end;
if rejs337 = 'Y' then do;
type1 = type1 + 1;
type = type + 1;
mess {type} = 'Type1:S337 Total NRFC days following last urg. reass
mismatch';
end;
end;

```

```

/*****
/* S338: TOTAL NRFC DAYS FOLLOWING LAST CLINICAL URGENCY INCREASE      */
/*      MISMATCH                                                       */
*/
/*      ADDED 1 JULY 2001                                             */
/*****/

      /***** S338 REMOVED FROM WAITING LIST *****/
if cdate > '31JUL2001'd then do;
  rejs338 = 'N';
if urgup_dt ne . then do;
  if pat_stat = 'R' and sr_date > urgup_dt and sr_date > reg_date and
    urgup_dt > reg_date and nrtot_ui = 0 then rejs338 = 'Y';

  if pat_stat = 'N' and ((rem_date ne .) or
    (rmreason in ('X','W','M','B','I','U','R','Z','Q','F','O','N','T')))
  then do;
    if urgup_dt >= x and sr_date >= x and urgup_dt < sr_date and
      (nrtot_ui < (nruiold + intck('day',sr_date,rem_date)))
    then rejs338 = 'Y';
    if urgup_dt >= x and sr_date >= x and urgup_dt > sr_date and
      (nrtot_ui < (nruiold + intck('day',urgup_dt,rem_date)))
    then rejs338 = 'Y';
    if urgup_dt < x and sr_date < x and
      (nrtot_ui < (nruiold + intck('day',x,rem_date)))
    then rejs338 = 'Y';
    if urgup_dt >= x and sr_date < x and
      (nrtot_ui < (nruiold + intck('day',urgup_dt,rem_date)))
    then rejs338 = 'Y';
    if urgup_dt < x and sr_date >= x and
      (nrtot_ui < (nruiold + intck('day',sr_date,rem_date)))
    then rejs338 = 'Y';
  end;

      /***** S338 REMAIN ON WAITING LIST *****/

if pat_stat = 'N' and (rem_date eq . or rmreason eq '') then do;
  if sr_date >= x and urgup_dt >= x and urg_date < sr_date and
    (nrtot_ui < (nruiold + intck('day',sr_date,cdate)+1))
  then rejs338 = 'Y';
  if sr_date >= x and urgup_dt >= x and urg_date > sr_date and
    (nrtot_ui < (nruiold + intck('day',urgup_dt,cdate)+1))
  then rejs338 = 'Y';
  if sr_date < x and urgup_dt < x and
    (nrtot_ui < (nruiold + intck('day',x,cdate)+1))
  then rejs338 = 'Y';
  if sr_date < x and urgup_dt >= x and
    (nrtot_ui < (nruiold + intck('day',urgup_dt,cdate)+1))
  then rejs338 = 'Y';
  if sr_date >= x and urgup_dt < x and
    (nrtot_ui < (nruiold + intck('day',sr_date,cdate)+1))
  then rejs338 = 'Y';
end;

```

```

        if rejs338 = 'Y' then do;
            type1 = type1 + 1;
            type = type + 1;
            mess {type} = 'Type1:S338 Total NRFC days following last urg. reass
mismatch';
        end;
    end;
end;

/**** END S338 ****/

if (type > 0) then output esis_iv.r&esisext.&campcode.;
if dset eq 'new' then output esis_iv.z&esisext.;

run;
/* ----- */
/* LOGGING INFORMATION FOR AHDW Job Control */
/* ----- */
data errcount (keep=rownum rowidstr Errmess issuecause Total);
set esis_iv.r&esisext.&campcode. (keep=campcode key urno row_number mess1-
mess20);

format rowidstr $50.;
rowidstr = campcode||key||urno;

rownum = row_number;
format issuecause $100.;
issuecause = 'N/A';
Total=1;

array mx {20}
    mess1-mess20 ;

do i=1 to 20 while (mx{i} ne '');
    errmess=substr(mx{i},7,4);
    output errcount;
end;

run;

data esis_iv.e&esisext.&campcode. /* file for all data sent */;
set esis_iv.z&esisext.;

format cdate date9. epid $15. ;

keep epid key--hi_post cdate campcode n_rec waitdays tot_wait
    w_stat birthdat overwait overadmt waitstat type type1 type2 type3;

format overwait overadmt waitstat w_stat $1. ;

output esis_iv.e&esisext.&campcode;
run;

```

```

proc sort data=esis_iv.r&esisext.&campcode.;
  by type1 type2 type3 mess1 mess2 mess3 ;
run;

PROC EXPORT DATA= esis_iv.e&esisext.&campcode.
  OUTFILE=
"F:\data\esis_data\esis_IV\filesdbf\e&esisext.&campcode..xls"
  DBMS=EXCEL97;
RUN;

/***** A Summary Report of Errors *****/

data ercount (keep=Errmess Total);
  set esis_iv.r&esisext.&campcode. (keep=mess1-mess10);

  Total=1;

  array mx {10} mess1-mess10;

  do i=1 to 10 while (mx{i} ne '');
    errmess=mx{i};
    output ercount;
  end;
run;

proc summary data=ercount nway missing;
  class errmess;
  var Total;
  output out=ercount1 (drop=_type_ _freq_) sum=;
run;

PROC EXPORT DATA= ercount1
  OUTFILE=
"f:\data\esis_data\esis_iv\filesdbf\m&esisext.&campcode..xls"
  DBMS=EXCEL97;
RUN;

/* ----- */
/* LOGGING INFORMATION FOR AHDW Job Control */
/* ----- */

data errexp (drop = mess11 -mess20);
set esis_iv.r&esisext.&campcode.;
run;

PROC EXPORT DATA= errexp
  OUTFILE=
"f:\data\esis_data\esis_iv\filesdbf\r&esisext.&campcode..xls"
  DBMS=EXCEL97;
RUN;

proc sql;

  create table experrs as
  SELECT rownum as ROW_NUMBER, rowidstr AS RECORD_ID_STR, Errmess as
ISSUE_CODE, issuecause as ISSUE_CAUSE
  FROM errcount;

quit;

```

```

PROC EXPORT DATA= WORK.Experss
      OUTFILE= "\\w023380\data\e&esisext.&campcode..txt"
      DBMS=TAB REPLACE;
RUN;

data load_tmpl;

FORMAT filename $255.;
FORMAT cdate $50.;
FORMAT rdate $50.;
FORMAT totalrec 4.0;
FORMAT passrec 4.0;
FORMAT warnrec 4.0;
FORMAT corrrec 4.0;
FORMAT rejrec 4.0;

run;

proc sql;

      CREATE TABLE load_stats like load_tmpl;

quit;

/*-----*/
/*      INSERT BLANK VALUES IN FOR UPDATING LATER      */
/*-----*/

proc sql;

      INSERT INTO load_stats VALUES ('','','1/1/2002', 0,0,0,0,0);

quit;

/*-----*/
/*      INSERT INDIVIDUAL VALUES INTO TABLE      */
/*-----*/
proc sql;

      UPDATE load_stats SET filename =
"f:\data\esis_data\esis_iv\rawdata\0002\&campcode.\D&esisext.&campcode..TXT";
      ALTER TABLE load_stats ADD bigstring VARCHAR(500);
quit;

```

```

proc sql;

    UPDATE load_stats SET cdate = "&CDATE.";
    UPDATE load_stats SET rdate = "&RDATE.";
    UPDATE load_stats SET totalrec = (SELECT sum(d_n_rec) FROM keep4later);
    UPDATE load_stats SET warnrec = (SELECT sum(Total) FROM ercount1 where
substr(errmess,1,5) eq 'Type3');
    UPDATE load_stats SET corrrec = (SELECT sum(Total) FROM ercount1 where
substr(errmess,1,5) eq 'Type2');
    UPDATE load_stats SET rejrec = (SELECT sum(Total) FROM ercount1 where
substr(errmess,1,5) eq 'Type1');
    UPDATE load_stats SET totalrec = 0 where totalrec = .;
    UPDATE load_stats SET warnrec = 0 where warnrec = .;
    UPDATE load_stats SET corrrec = 0 where corrrec = .;
    UPDATE load_stats SET rejrec = 0 where rejrec = .;
    UPDATE load_stats SET passrec = totalrec - (warnrec + corrrec + rejrec);

quit;

data tmp1;

FORMAT bigstring $1000.;
FORMAT filename $255.;
FORMAT cdate $50.;
FORMAT rdate $50.;
FORMAT totalrec $4.0;
FORMAT passrec $4.0;
FORMAT warnrec $4.0;
FORMAT corrrec $4.0;
FORMAT rejrec $4.0;

run;

data tmp1;
set load_stats;

FORMAT cttotalrec $4.0;
FORMAT cpassrec $4.0;
FORMAT cwarnrec $4.0;
FORMAT ccorrrec $4.0;
FORMAT crejrec $4.0;

cttotalrec = totalrec;
cpassrec = passrec;
ccorrrec = corrrec;
cwarnrec = warnrec;
crejrec = rejrec;

run;

```

```

proc sql;
    UPDATE tmp1 SET bigstring =
'f:\data\esis_data\esis_iv\Programs\SubLogger.exe';
    UPDATE tmp1 SET bigstring = trim(bigstring)||' '||trim(filename)||' '
' || "&CAMPCODE.";
    UPDATE tmp1 SET bigstring = trim(bigstring)||' '||trim(cdate)||' ';
    UPDATE tmp1 SET bigstring = trim(bigstring)||' '||trim(rdate)||' ';
    UPDATE tmp1 SET bigstring = trim(bigstring)||'
' || compress(trim(ctotalrec));
    UPDATE tmp1 SET bigstring = trim(bigstring)||' ' || compress(trim(cpassrec));
    UPDATE tmp1 SET bigstring = trim(bigstring)||'
' || compress(trim(cwarnrec));
    UPDATE tmp1 SET bigstring = trim(bigstring)||'
' || compress(trim(ccorrrec));
    UPDATE tmp1 SET bigstring = trim(bigstring)||' ' || compress(trim(crejrec));
    UPDATE tmp1 SET bigstring = trim(bigstring)||' ' || trim(rdate)||' ';
    UPDATE tmp1 SET bigstring = trim(bigstring)||'
' || "\w023380\data\e&esisext.&campcode..txt";
    UPDATE tmp1 SET bigstring = trim(bigstring)||' ' || 'The Sas
System\mainload_allmonths_fix.sas';

quit;

/*-----*/
/*      SET THE VALUE VARIABLES                               */
/*-----*/

%LET ARGSTRING = '';

data _null_;
set tmp1;

CALL SYMPUT('ARGSTRING',bigstring);

run;

systask command "&ARGSTRING.";

```