```
/*Assigned library name reference, annoCRF, to folder containing xlsx file
of annotated eCRF.*/
libname annoCRF '/home/dimplepatelch0/5457c/annoCRF';

/*Imported xlsx file version of annotated CRF. Used Adobe Acrobat DC to convert
the original PDF file into an xlsx file format.*/
/* Generated Code (IMPORT) */
/* Source File: eCRFs_annotated.xlsx */
/* Source Path: /home/dimplepatelch0/5457c/annoCRF */
/* Code generated on: 8/5/19, 10:25 AM */

%web_drop_table(WORK.IMPORT);
filename ecrf1 '/home/dimplepatelch0/5457c/annoCRF/eCRFs_annotated.xlsx';

proc import datafile=ecrf1 dbms=xlsx out=annocrf.ecrf1;
    getnames=yes;
run;

/*Generated table, row, column, and data attributes with PROC Contents. */
proc contents data=annocrf.ecrf1;
run;

/*Printed SAS table version of imported annotated eCRF xlsx file. */
/* proc print data=annocrf.ecrf1; */
/* run; */
/*Cleaned up the dataset of the imported annotated eCRF by deleting
the first 4 and last 20 rows. Then the first column's non-SAS-compatible
label was renamed to num101 for easy future data manipulation.*/
data annocrf.ecrf3;
    set annocrf.ecrf1(rename=("eCRF - ANNOTATED for CMP5457C"n=num101)) nobs=num;
    if num - _n_ < 20 then
        delete;
    if _n_=<  4 then
        delete;
run;

/*Uses PERL language with 2 functions, prxparse & prxmatch,
to locate any row with the phrase "DOMAINS:", which is
on each CRF page that contains CRF variables paired up
with an SDTM variable.  */
data annocrf.perlparse;
    set annocrf.ecrf3;
    if _n_=1 then
    pattern_num=prxparse("/DOMAIN?/");
    retain pattern_num;
    position=prxmatch(pattern_num, num101);
    file print;
    put pattern_num=num101=position=;
    retain domainloc domainnaml;
    if position ne 0 then
        do;
            domainloc=_n_;
        end;
run;

data _null_;
    if _n_=1 then
        do;
            if 0 then
            set annocrf.perlparse;
            declare hash split(dataset:"annocrf.perlparse", multidata:'y');
            split.definekey('domainloc');
            split.definedata(all:'y');
            split.definedone();
        end;
    do until(last.domainloc);
        set annocrf.perlparse;
        by domainloc;
        split.add();
    end;
    split.output(dataset: 'CRFSPLIT' || put(domainloc, best.-l));
    split.clear();
run;

proc sql;
    select libname, memname into :nam from dictionary.tables
```

```
77      where memname ? "CRFSPLIT";
78  quit;
79
80  /* WORKS FOR CRF8   regID=prxparse('s/(?:.*\s?\s?DOMAIN.[:])\s(\w\w).*\s(\w\w)/$1 $2/'); */
81  data work.c1031;
82  set work.crfsplit8;
83      length Domains PageNo MapTerm $100;
84      regID=prxparse('s/(?:.*\s?\s?DOMAIN.[:])\s(\w\w).*\s(\w\w)/$1 $2/o');
85      rege=prxparse('s/(?:.*(OID\:)\s)(\w{4}\d)(?:\s*.*)/$2/o');
86      regf=prxparse('s/([A-Z]{6}\s[\=]\s.*\s).([A-Z]{7}\s[\=]\s(\w{6})\s)+.*([A-Z]{5}\s[\=]\s.*)\s.([A-Z]{5}\s[\=]\s[A-Z]
87      if prxmatch(regID,num101) ne 0 then do;
88      Domains=prxchange(regID,-1,num101);
89      end;
90      if prxmatch(rege,num101) ne 0 then do;
91      PageNo=prxchange(rege,-1,num101);
92      end;
93      if prxmatch(regf,num101) ne 0 then do;
94      MapTerm=prxchange(regf,-1,num101);
95      end;
96  run;
97
98  proc print data=work.crfsplit8;
99  run;
100
101 proc print data=work.c1031;
102 run;
103
```