

Providing Extended Help in FSEDIT Sessions

Paul Hamilton, Fred Hutchinson Cancer Research Center

When entering data directly into the SAS System via the FSEDIT procedure in SAS/FSP®, what type of information is expected in a particular variable may not be clear. Proc FSEDIT does provide a field help function (press F1 while the cursor is on a field), but this does not always provide sufficient information. For example, the system may respond that the name of the field is GENDER, the field is character, and has a format of \$GENFMT. However, only the application developer will know if the data expected is Male/Female, M/F, or how missing values should be entered. This paper presents a method of providing enhanced help which simplifies data entry.

The necessary steps are to define a key sequence in the FSEDIT application which invokes the special help function (e.g. **FieldHelp** to shift-F1), and add some Screen Control Language code to the application (provided below). During data entry, one may obtain native help by pressing F1, or extended help by pressing shift-F1.

The SCL code follows on the next page. Numbers in this article reference numbers in the code. Three assumptions are made: (1) formats are stored in LIBRARY.FORMATS; (2) formats are defined for all coded variables; and (3) descriptive labels for each variable are provided in the SAS data set. Also, this techniques will not work if the FSEDIT session is initiated by submitting a PROC FSEDIT; invocation from a command line or SCL CALL FSEDIT is required.

Most of the code is generic, not changed between data sets. Only three places must be altered for use in each new FSEDIT screen: ①, ②, and ④. Sections ① and ② are separate lists of the numeric and character variables in the current dataset. These lists may be typed into the code by hand, or a separate application may be used to generate them (contact the author for details). Section ④ is simply the name of the current dataset; there is currently no built-in function which provides this infor-

mation automatically.

When the shift-F1 key is pressed, the SCL in the Main section recognizes a special user command (**FieldHelp**). Section ③ looks for a specific dataset, Work._FmtLib, and creates it if necessary. This contains all the format specifications in dataset form. Section ⑤ determines the name, label, and format of the field the cursor was on when the FieldHelp key was pressed. The name of the format field in the dataset contains characters which are inconvenient, so section ⑥ strips them out.

The code starting in section ⑦ performs the real work. Numeric and character variables are managed separately, but the techniques are parallel in form and usage. More exceptions will occur while processing numeric variables. First, the developer might wish to provide special feedback (⑧) for particular formats such as dates, Yes/No, and other common data usage patterns occurring in the application. If the current variable is not associated with one of these formats, the code then looks up (⑨) the format from the code table generated in section ③. A dialog box pops up showing all legal values for the field (taken from the format library), permitting easy selection by mouse (when a GUI is available) or arrow keys (for the mainframe). The CALL WREGION commands could be modified to suit the screen size and resolution for different environments. If no format is used, just the variable label is provided as help text.

This code could be customized for different environments, user needs, data types, and hardware assets. Once modified, it can be quickly copied into new FSEDIT SCL screens, and provides helpful and consistent feedback during data entry.

SAS and SAS/FSP are registered trademarks of SAS Institute, Inc. in the USA and other countries. ® indicates USA registration.

Contact the author at: hamilton.hiv-net.fhcrc.org, 1124 Columbia St., MP-900 Seattle, WA 98104

Providing Extended Help in FSEDIT Sessions.

```

* Array of Numeric variables;                                ①
  array numf(*)      AGE;
  array numn(*) $ 8 ('AGE');

* Array of Character variables;                               ②
  array charf(*)      NAME  GENDER  MARSTAT;
  array charn(*) $ 8 ('NAME' 'GENDER' 'MARSTAT');

FSEINIT:
  control enter;
return;

INIT:
return;

MAIN:
  if word(1,'U') eq 'FIELDHELP' then do;
    rc = exist('Work._FmtLib','DATA');
    if rc eq 0 then do;                                     ③
      submit continue;
      proc format library = library
        cntlout = _fmtlib;
      run;
    endsubmit;
  end;

  thisds= open('PNWSUG.EXAMPLE', 'I');
  fdsid = open('Work._FmtLib', 'I');
  field = curfld();                                         ⑤
  ftype = vartype(thisds,varnum(thisds,field));
  thisfmt = varfmt(thisds,varnum(thisds,field));
  thislab = varlabel(thisds,
    varnum(thisds, field));

  dot = index(thisfmt, '.');                                ⑥
  if (dot ne 0) then do;
    thisfmt = substr(thisfmt, 1, dot-1);

```

```

    lastchar=substr(thisfmt,length(thisfmt),1);
    if indexc(lastchar, '1234567890') ne 0 then
      thisfmt=substr(thisfmt,1,length(thisfmt)-1);
    lastchar=substr(thisfmt,length(thisfmt), 1);
    if indexc(lastchar, '1234567890') ne 0 then
      thisfmt=substr(thisfmt,1,length(thisfmt)-1);
  end; * Processing dots;

  select(ftype);                                           ⑦
    when('N') do;
      select(thisfmt);                                     ⑧
        when('MDDYY') _msg_ =
          'Enter a date value for ' || thislab;
        when('CHKBLNK') _msg_ = '1=X
          2=(blank) for ' || thislab;
        when('YESNO') _msg_ =
          'Enter Y or N for ' || thislab;
        otherwise do;
          submit continue;                                ⑨
          data _fmt;
            set _fmtlib;
            where fmtname eq '&thisfmt';
            value = input(start, 16.);
            keep start value label;
          run;
        endsubmit;
        call wregion(5, 5, 15, 60);
        newdsid = open('Work._Fmt', 'I');
        nob = attrn(newdsid, 'Any');
        if (nob eq 0) then do;
          rc = close(newdsid);
          rc = delete('Work._Fmt');
          _msg_ = 'Numeric Field: ' ||
            thislab;
        return;
        end;
        do i = 1 to dim(numf);
          if (numn[i] eq field) then do;
            numf[i] = datalistn(newdsid,

```

Providing Extended Help in FSEDIT Sessions.

```

        'Value Label');
        end;
        end;
        rc = close newdsid);
        rc = delete('Work._Fmt');
    end;
end; * Processing select formats;
end; * Processing Numeric variable;
when('C') do;
    thisfmt = compress(thisfmt, '$');
    submit continue;
    data _fmt;
        set _fmtlib;
        where fmtname eq '&thisfmt';
        keep start label;
    run;
endsubmit;
newdsid = open('Work._Fmt', 'I');
nobs = attrn(newdsid, 'Any');
if (nobs eq 0) then do;
    rc = close(newdsid);
    rc = delete('Work._Fmt');
    _msg_ = 'Character field: ' ||
        thislab;
end;
if nobs eq 0 then leave;
call wregion(5, 5, 15, 60);
do i = 1 to dim(charf);
    if charn[i] eq field then
        charf[i] = datalistc(newdsid,
            'Start Label');
end;
rc = close(newdsid);
rc = delete('Work._Fmt');
end; * Processing character variables;
end; * Select on FTYPE;
rc = close(thisds);
rc = close(fdsid);
call nextcmd();

```

```

        end; * Processing FieldHelp;
return;

TERM:
return;

FSETERM:
    if (exist('Work._FmtLib'))
        then rc = delete('Work._FmtLib');
return;

```