MIMO – Multiple Inputs Multiple Outputs
Multiplying the Fun while doing the DATA step

John Fahey
Reproductive Care Program of Nova Scotia
Halifax, NS, Canada

Abstract

Every SAS® programmer has done the DATA step, one foot at a time:

```
Data  Dance;                    Coders' Corner
Infile Do_Si_Do;                2006-09-18T09:24-04:00
Input OneFoot &TheOther;
Run; /* back to your partner */
```

Or

```
Data GetUp;
Set BackDown;
Run; /* off at the mouth */
```

Sometimes the best way to arrange your dance card is with multiple partners.

```
title1  'Demonstrate how to read in a single input file';
title2 'and create a single output dataset';
filename    MBAStats    'c:\temp\MBA.txt'
;   /* reformatted form of this file from 20060609          */
/* http://www.mbans.ca/league.php?
     scriptName=LEAGUEINFO&leagueID=8439&leagueInfoID=29583 */

data    Work.MBA_Demo;
    infile  MBAStats
        delimiter   =   '09'x   /* tab                      */
        firstobs    =   2       /* skip header              */
        truncover;  /* blanks at end-of-record to missing   */
    input   Rank
            Name : $20.         /* read up to 20 chars or   */
            Team : $10.         /* tab, whichever is first  */
            Games_Played
            Points_Scored
            Points_per_Game
            Free_Throws_Attempted
            Free_Throws_Made
            Free_Throw_Proportion;
    format  Free_Throw_Proportion Percent7.1;
run;
```

```
title1  'Now a separate dataset for each team -';
title2  'i.e. single input, multiple outputs';
data    Work.Cole_Harbour
        Work.Fairview
        Work.Sackville
        Work.Timberlea;
    infile  MBAStats delimiter = '09'x firstobs = 2 truncover;
    input   Rank Name : $20. Team : $10. Games_Played Points_Scored
            Points_per_Game Free_Throws_Attempted Free_Throws_Made
            Free_Throw_Proportion;
    format  Free_Throw_Proportion Percent7.1;
    select  ( Team );
        when    ( 'CH' )        output  Cole_Harbour;
        when    ( 'Fairview' )  output  Fairview;
        when    ( 'Sackville' ) output  Sackville;
        when    ( 'Timberlea' ) output  Timberlea;
        otherwise   put 'UNKNOWN TEAM';
    end;    /* select */
run;
```

```
title1  'Or the reverse, multiple input files with a single output';
title2  'using Philadelphia, PA (daily) climate normals';

filename
    TemPrecp    /* Temperature and Precipitation formatted file    */
    url
'http://hurricane.ncdc.noaa.gov/climatenormals/clim84/PA/PA366889.txt';

filename
    Snowfall    /* use ASCII (unformatted) file                    */
    url 'http://hurricane.ncdc.noaa.gov/climatenormals/
                                    clim20-02/NWS_SNOW_MNFALL_dly.dat';

data    Work.PhillyWx
    ( drop  =   EltNum );   /* ditch loop counter                  */
    array   Elements { * } MinTemp MaxTemp AveTemp
                           HtgDDays ClgDDays TPrecip Snow;
    format  WthrDate YYMMDD10.;
        infile  TemPrecp    /* 6 elements - temp, precipitation    */
            firstobs    =   12; /* skip header                     */
        infile  SnowFall
            firstobs    =   347 /* move down to station # 366889   */
            lrecl       =   1102;   /* whole year on one line      */
```

```
    do WthrDate = '01Jan2006'D to '31Dec2006'D  /* use this year as*/
;       /* storage area for 30-year average; neither has Feb. 29th */
        infile  TemPrecp    /* all elements accessible at once in  */
            n   =   86;     /* multi-line input buffer             */
        do  EltNum = 1 to 6;/* first file has six elements         */
            input   #(17*(EltNum-1)+month(WthrDate)) /* line # */
                    @(3*day(WthrDate)+2)
                Elements { EltNum } 3. @@;
        end; /* do EltNum - N.B. has value 7 after loop            */
        infile  Snowfall;   /* separate element - mean daily .1 in */
        input   @(3*mod(juldate(WthrDate),1000)+5)
                  Elements { EltNum } 3. @@;
        do  EltNum = 1 to 7;
            if ( Elements { EltNum } in ( -88, -99 ) ) then
                Elements { EltNum } = . ;
        end;    /* recode missing values                           */
        output;
    end;    /* do WthrDate                                         */
    stop;   /* after all 365 days read in; avoids "LOST CARD"      */
run;
```

```
    infile  ChpBlock
        End     =   End_Chp_Block
        LRecL   =   303
        TruncOver; /* … */
 do while ( not End_Chp_Block );
        input   Chp_or_Block $ 1-7  /* for chapters, */
             Short_Lab 8-47         /* have rubric ranges appended */
             Long_Lab 48-303;/* always have rubric ranges appended */

…
if ( AnyAlpha ( Chp_or_Block ) ) then do;/* Block Label */
            Block   =   Chp_or_Block;
            Short_Label =   Short_Lab;
            if (First_Rubric = ' ') then do;/*degenerate,see above*/
                First_Rubric    =   substr ( Chp_or_Block, 1, 3 );
                Last_Rubric     =   substr ( Chp_or_Block, 5, 3 );
            end;
            output  Work.ICD_Block_Labels;
        end; /* if */
        else do;                                    /* Chapter Label */
            Chapter =   input ( Chp_or_Block, 2. ); /* a numeric*/
            /* usually expressed in Roman numerals (see format) */
            Short_Posn  =   prxmatch ( R_R_Expression, Short_Lab ); /*…*/
            output  Work.ICD_Chapter_Labels;
        end;    /* else */
    end;    /* while not end-of-first-file */
```

```
infile  Category
        End     =   End_Rubric
        LRecL   =   303
        TruncOver;
    do while ( not End_Rubric );
        input   Category 1-7
                Short_Label 8-47
                Long_Label 48-303;
        output  Work.ICD_Category_Labels;
    end;    /* while not end-of-second-file */
run;
```