Chapter 10
The DATASOURCE Procedure

Chapter Table of Contents

Chapter 10
The DATASOURCE Procedure

Overview

The DATASOURCE procedure extracts time series data from many different kinds of data files distributed by various data vendors and stores them in a SAS data set. Once stored in a SAS data set, the time series variables can be processed by other SAS procedures.

The DATASOURCE procedure has statements and options to extract only a subset of time series data from an input data file. It gives you control over the frequency of data to be extracted, time series variables to be selected, cross sections to be included, and the time range of data to be output.

The DATASOURCE procedure can create auxiliary data sets containing descriptive information on the time series variables and cross sections. More specifically, the OUTCONT= data set contains information on time series variables, the OUTBY= data set reports information on the cross-sectional variables, and the OUTALL= data set combines both time series variables and cross-sectional information.

The output variables in the output and auxiliary data sets can be assigned various attributes by the DATASOURCE procedure. These attributes are labels, formats, new names, and lengths. While the first three attributes in this list are used to enhance the output, the length attribute is used to control the memory and disk-space usage of the DATASOURCE procedure.

Data files currently supported by the DATASOURCE procedure include

- U.S. Bureau of Economic Analysis data files:
  - National Income and Product Accounts tapes
  - National Income and Product Accounts diskettes
  - S-page diskettes
- U.S. Bureau of Labor Statistics data files:
  - Consumer Price Index Surveys
  - Producer Price Index Survey
  - National Employment, Hours, and Earnings Survey
  - State and Area Employment, Hours, and Earnings Survey
- Standard & Poor's Compustat Services Financial Database Files:
  - COMPUSTAT Annual
  - COMPUSTAT 48 Quarter

- Center for Research in Security Prices (CRSP) data files:
  - Daily Binary Format Files
  - Monthly Binary Format Files
  - Daily Character Format Files
  - Monthly Character Format Files
  - Daily IBM Binary Format Files
  - Monthly IBM Binary Format Files
  - 1995 CDROM Character Format Files
  - 1995 CDROM UNIX (SUN) Binary Format Files
  - ACCESS97 CDROM Binary Format Files
- DRI/McGraw-Hill data files:
  - Basic Economics Data (formerly CITIBASE)
  - DRI Data Delivery Service files
  - Tape Format CITIBASE Data Files
  - DRI Data Delivery Service Time Series
  - PC Diskette format CITIBASE Databases
- FAME Information Services Databases
- Haver Analytics data files
- International Monetary Fund's Economic Information System data files:
  - International Financial Statistics
  - Direction of Trade Statistics
  - Balance of Payment Statistics
  - Government Finance Statistics
- Organization for Economic Cooperation and Development:
  - Annual National Accounts
  - Quarterly National Accounts
  - Main Economic Indicators

Getting Started

Structure of a SAS Data Set Containing Time Series Data

SAS procedures require time series data to be in a specific form recognizable by the SAS System. This form is a two-dimensional array, called a SAS data set, whose columns correspond to series variables and whose rows correspond to measurements of these variables at certain time periods.

The time periods at which observations are recorded can be included in the data set as a time ID variable. The DATASOURCE procedure does include a time ID variable by the name of DATE.

For example, the following data set, extracted from a CITIBASE data file, gives the foreign exchange rates for Japan, Switzerland, and the United Kingdom, respectively.

Time ID variable	Time Series Variables		
DATE	EXRJAN	EXRSW	EXRUK
SEP1987	143.290	1.50290	164.460
OCT1987	143.320	1.49400	166.200
NOV1987	135.400	1.38250	177.540
DEC1987	128.240	1.33040	182.880
JAN1988	127.690	1.34660	180.090
FEB1988	129.170	1.39160	175.820

Figure 10.1. The Form of SAS Data Sets Required by Most SAS/ETS Procedures

Reading Data Files

The DATASOURCE procedure is designed to read data from many different files and to place them in a SAS data set. For example, if you have a DRI Basic Economics data file you want to read, use the following statements:

```
proc datasource filetype=dribasic infile=citifile out=dataset;
run;
```

Here, the FILETYPE= option indicates that you want to read DRI's Basic Economics data file, the INFILE= option specifies the fileref CITIFILE of the external file you want to read, and the OUT= option names the SAS data set to contain the time series data.

Subsetting Input Data Files

When only a subset of a data file is needed, it is inefficient to extract all the data and then subset it in a subsequent DATA step. Instead, you can use the DATASOURCE procedure options and statements to extract only needed information from the data file.

The DATASOURCE procedure offers the following subsetting capabilities:

- the INTERVAL= option controls the frequency of data output
- the KEEP or DROP statements selects a subset of time series variables
- the RANGE statement restricts the time range of data
- the WHERE statement selects a subset of cross sections

Controlling the Frequency of Data – The INTERVAL= Option

The OUT= data set can only contain data with the same frequency. If the data file you want to read contains time series data with several frequencies, you can indicate the frequency of data you want to extract with the INTERVAL= option. For example, the following statements extract all monthly time series from the DRIBASIC file CITIFILE:

```
proc datasource filetype=dribasic infile=citifile
                interval=month  out=dataset;
run;
```

When the INTERVAL= option is not given, the default frequency defined for the FILETYPE= type file is used. For example, the statements in the previous section extract yearly series since INTERVAL=YEAR is the default frequency for DRI's Basic Economic Data files.

To extract data for several frequencies, you need to execute the DATASOURCE procedure once for each frequency.

Selecting Time Series Variables – The KEEP and DROP Statements

If you want to include specific series in the OUT= data set, list them in a KEEP statement. If, on the other hand, you want to exclude some variables from the OUT= data set, list them in a DROP statement. For example, the following statements extract monthly foreign exchange rates for Japan (EXRJAN), Switzerland (EXRSW), and the United Kingdom (EXRUK) from a DRIBASIC file CITIFILE:

```
proc datasource filetype=dribasic infile=citifile
                interval=month  out=dataset;
   keep  exrjan exrsw exruk;
run;
```

The KEEP statement also allows input names to be quoted strings. If the name of a series in the input file contains blanks or special characters that are not valid SAS name syntax, put the series name in quotes to select it. Another way to allow the use of special characters in your SAS variable names, is to use the SAS options statement to designate VALIDVARNAME= ANY. This option will allow PROC DATASOURCE to include special characters in your SAS variable names. The following is an example of extracting series from a FAME database using the DATASOURCE procedure.

```
proc datasource filetype=fame dbname='fame_nys /disk1/prc/prc'
                interval=weekday out=outds outcont=attrds;
   range '1jan90'd to '1feb90'd;
   keep cci.close
        '{ibm.high,ibm.low,ibm.close}'
        'mave(ibm.close,30)'
        'crosslist({gm,f,c},{volume})'
        'cci.close+ibm.close';
   rename 'mave(ibm.close,30)' = ibm30day
          'cci.close+ibm.close' = cci_ibm;
run;
```

The resulting output data set OUTDS contains the following series: DATE, CCI_CLOS, IBM_HIGH, IBM_LOW, IBM_CLOS, IBM30DAY, GM_VOLUM, F_VOLUME, C_VOLUME, CCI_IBM.

Obviously, to be able to use KEEP and DROP statements, you need to know the name of time series variables available in the data file. The OUTCONT= option gives you this information. More specifically, the OUTCONT= option creates a data set containing descriptive information on the same frequency time series. This descriptive information includes series names, a flag indicating if the series is selected for output, series variable types, lengths, position of series in the OUT= data set, labels, format names, format lengths, format decimals, and a set of FILETYPE= specific descriptor variables. For example, the following statements list some of the monthly series available in the CITIFILE:

```
filename citifile 'host-specific-file-name' <host-options>;
proc datasource filetype=dribasic infile=citifile
                interval=month  outcont=vars;
run;

title1 'Some Time Series Variables Available in CITIFILE';
proc print data=vars noobs;
run;
```

Some Time Series Variables Available in CITIFILE

name	select	type	length	varnum	label
EXRJAN	1	1	5	.	FOREIGN EXCHANGE RATE: JAPAN (YEN PER U.
EXRSW	1	1	5	.	FOREIGN EXCHANGE RATE: SWITZERLAND (SWIS
EXRUK	1	1	5	.	FOREIGN EXCHANGE RATE: UNITED KINGDOM (C

description	format	ffmatl	fformatd	code
FOREIGN EXCHANGE RATE: JAPAN (YEN PER U.S.$)		0	0	EXRJAN
FOREIGN EXCHANGE RATE: SWITZERLAND (SWISS FRANC PER U.S.$)		0	0	EXRSW
FOREIGN EXCHANGE RATE: UNITED KINGDOM (CENTS PER POUND)		0	0	EXRUK

Figure 10.2. Partial Listing of the OUTCONT= Data Set

Controlling the Time Range of Data – The RANGE Statement

The RANGE statement is used to control the time range of observations included in the output data set. For example, if you want to extract the foreign exchange rates from September, 1987 to February, 1988, you can use the following statements:

```
filename citifile 'host-specific-file-name' <host-options>;
proc datasource filetype=dribasic infile=citifile
                interval=month  out=dataset;
   keep  exrjan exrsw exruk;
   range from 1987:9 to 1988:2;
run;

title1 'Printout of the OUT= Data Set';
proc print data=dataset noobs;
run;
```

Printout of the OUT= Data Set

date	exrjan	exrsw	exruk
SEP1987	143.290	1.50290	164.460
OCT1987	143.320	1.49400	166.200
NOV1987	135.400	1.38250	177.540
DEC1987	128.240	1.33040	182.880
JAN1988	127.690	1.34660	180.090
FEB1988	129.170	1.39160	175.820

Figure 10.3. Subset Obtained by KEEP and RANGE Statements

Reading in Data Files Containing Cross Sections

Some data files group time series data with respect to cross-section identifiers; for example, International Financial Statistics files, distributed by IMF, group data with respect to countries (COUNTRY). Within each country, data are further grouped by Control Source Code (CSC), Partner Country Code (PARTNER), and Version Code (VERSION).

If a data file contains cross-section identifiers, the DATASOURCE procedure adds them to the output data set as BY variables. For example, the data set in Table 10.1 contains three cross sections:

- the first one is identified by (COUNTRY='112' CSC='F' PARTNER=' ' VERSION='Z')
- the second one is identified by (COUNTRY='146' CSC='F' PARTNER=' ' VERSION='Z')
- the third one is identified by (COUNTRY='158' CSC='F' PARTNER=' ' VERSION='Z').

Table 10.1. The Form of a SAS Data Set Containing BY Variables

BY Variables				Time ID Variable	Time Series Variables	
COUNTRY	CSC	PARTNER	VERSION	DATE	EFFEXR	EXRINDEX
112	F		Z	SEP1987	9326	12685
112	F		Z	OCT1987	9393	12813
112	F		Z	NOV1987	9626	13694
112	F		Z	DEC1987	9675	14099
112	F		Z	JAN1988	9581	13910
112	F		Z	FEB1988	9493	13549
146	F		Z	SEP1987	12046	16192
146	F		Z	OCT1987	12067	16266
146	F		Z	NOV1987	12558	17596
146	F		Z	DEC1987	12759	18301
146	F		Z	JAN1988	12642	18082
146	F		Z	FEB1988	12409	17470
158	F		Z	SEP1987	13841	16558
158	F		Z	OCT1987	13754	16499
158	F		Z	NOV1987	14222	17505
158	F		Z	DEC1987	14768	18423
158	F		Z	JAN1988	14933	18565
158	F		Z	FEB1988	14915	18331

Note that the data sets in Figure 10.1 and Table 10.1 are two different ways of representing the same data, namely foreign exchange rates for three different countries: the United Kingdom (COUNTRY='112'), Switzerland (COUNTRY='146') and Japan (COUNTRY='158'). The first representation (Figure 10.1) incorporates country

names into the series names, while the second representation (Table 10.1) represents countries as different cross sections. See “Time Series and SAS Data Sets” in Chapter 2, “Working with Time Series Data.”

Obtaining Descriptive Information on Cross Sections

If you want to know the unique set of values BY variables assume for each cross section in the data file, use the OUTBY= option. For example, the following statements list some of the cross sections available for an IFS file.

```
filename ifsfile 'host-specific-file-name' <host-options>;
proc datasource filetype=imfifsp infile=ifsfile
                interval=month outby=xsection;
run;

title1 'Some Cross Sections Available in IFSFILE';
proc print data=xsection noobs;
run;
```

Some Cross Sections Available in IFSFILE

country	csc	partner	version	start_date	end_date	ntime	nobs	nseries	nselect	nsse	ncase	name
1	F	900	Z	.	.	.	0	0	0			WORLD
1	F		Z	JAN1957	DEC1989	396	396	46	23			WORLD
1	T		Z	JAN1957	DEC1989	396	396	16	8			WORLD
10	F		Z	JAN1957	DEC1989	396	396	32	16			ALL COUNTRIES
10	F	900	Z	.	.	.	0	0	0			ALL COUNTRIES
10	M		Z	JAN1957	NOV1989	395	395	2	1			ALL COUNTRIES
10	T		Z	JAN1957	DEC1989	396	396	18	9			ALL COUNTRIES
16	F		Z	JAN1970	SEP1989	237	237	12	6			OFFSHORE BNKING CTRS
16	F	900	Z	.	.	.	0	0	0			OFFSHORE BNKING CTRS
24	F		Z	JAN1962	JUL1989	331	331	2	1			ACP COUNTRIES

Figure 10.4. Partial Listing of the OUTBY= Data Set

The OUTBY= data set reports the total number of series, NSERIES, defined in each cross section, NSELECT of which represent the selected variables. If you want to see the descriptive information on each of these NSELECT variables for each cross section, specify the OUTALL= option. For example, the following statements print descriptive information on the eight series defined for cross section (COUNTRY='1' CSC='T' PARTNER=' ' and VERSION='Z'):

```
filename ifsfile 'host-specific-file-name' <host-options>;
proc datasource filetype=imfifsp infile=ifsfile interval=month
                outall=ifsall;
run;
```

```
title1 'Time Series Defined in Cross Section';
title2 "COUNTRY='1'  CSC='T'  PARTNER=' '  VERSION='Z'";
proc print data=ifsall noobs;
   where country='1' and csc='T' and  partner=' ' and version='Z';
run;
```

```
                    Time Series Defined in Cross Section
               COUNTRY='1'  CSC='T'  PARTNER=' '  VERSION='Z'

                       s
c     p  v             e                                                 f
o     a  e             l     l  v   b                                 f  o
u     r  r             e     e  a   l  l                              o  r
n     t  s    n     k  c  t  n  r   k  a                              r  m
t  c  n  i    a     e  t  y  g  n   n  b                              m  a
r  s  e  o    m     p  e  p  t  u   u  e                              a  t
y  c  r  n    e     t  d  e  h  m   m  l                              t  l

1  T     Z  F__2KS  1  1  1  5  .  26  TOTAL PURCHASES                   0
1  T     Z  F__2LA  1  1  1  5  .  27  REPMTS.BY REPUR.IN PERIOD         0
1  T     Z  F__2MS  1  1  1  5  .  28  TOTAL PURCHASES BY OTHERS         0
1  T     Z  F__2NS  1  1  1  5  .  29  TOTAL REPURCHASES BY OTHERS       0
1  T     Z  F_C2KS  1  1  1  5  .  30  TOTAL PURCHASES,CUM.              0
1  T     Z  F_C2LA  1  1  1  5  .  31  REPAYMENTS BY REPURCHASE,CUM.     0
1  T     Z  F_C2MS  1  1  1  5  .  32  TOTAL PURCHASES BY OTHERS,CUM     0
1  T     Z  F_C2NS  1  1  1  5  .  33  TOTAL REP.BY OTHERS,CUM.          0

                  e                         d                          b
f        s        n                   s     a  d         d             a
o        t        d              c    u  s  t  u         u             s  s
r        _        _   n          _    b  c  a  _         _             e  o
m        d        d   t    n     n    j  d  t  c         n          n  y  u
a        a        a   i    o     a    e  a  y  o         a          d  e  r
t        t        t   m    b     m    c  t  p  d         m          e  a  c
d        e        e   e    s     e    t  a  e  e         e          c  r  e

0  JAN1957  DEC1989  396  396  WORLD        F  S  MILLIONS OF SDRS  1     T
0  JAN1957  DEC1989  396  396  WORLD        F  S  MILLIONS OF SDRS  2     T
0  JAN1957  DEC1989  396  396  WORLD        F  S  MILLIONS OF SDRS  1     T
0  JAN1957  DEC1989  396  396  WORLD        F  S  MILLIONS OF SDRS  2     T
0  JAN1957  NOV1986  359  359  WORLD     C  S  S  MILLIONS OF SDRS  1
0  JAN1957  DEC1989  396  396  WORLD     C  S  S  MILLIONS OF SDRS  1
0  JAN1957  NOV1986  359  359  WORLD     C  S  S  MILLIONS OF SDRS  1
0  JAN1957  DEC1989  396  396  WORLD     C  S  S  MILLIONS OF SDRS  1
```

Figure 10.5. Partial Listing of the OUTALL= Data Set

The OUTCONT= data set contains one observation for each time series variable with the descriptive information summarized over BY groups. When the data file contains no cross sections, the OUTCONT= and OUTALL= data sets are equivalent, except that the OUTALL= data set also reports time ranges for which data are available. The OUTBY= data set in this case contains a single observation reporting the number of series and time ranges for the whole data file.

Subsetting a Data File Containing Cross Sections

Data files containing cross sections can be subsetted by controlling which cross sections to include in the output data set. Selecting a subset of cross sections is accomplished using the WHERE statement. The WHERE statement gives a condi-

tion the BY variables must satisfy for a cross section to be selected. For example, the following statements extract the monthly effective exchange rate (F_X_AM) and exchange rate index (F_X_AF) for the United Kingdom (COUNTRY='112'), Switzerland (COUNTRY='146'), and Japan (COUNTRY='158') for the period from September, 1987 to February, 1988.

```
filename ifsfile 'host-specific-file-name' <host-options>;
proc datasource filetype=imfifsp infile=ifsfile interval=month
                out=exchange;
   where country in ('112','146','158') and partner=' ';
   keep  f_x_ah f_x_am;
   range from '01sep87'd to '01feb88'd;
run;

title1 'Printout of the OUT= Data Set';
proc print data=exchange noobs;
run;
```

Renaming Time Series Variables

Sometimes the time series variable names as given by data vendors are not descriptive enough, or you may prefer a different naming convention. In such cases, you can use the RENAME statement to assign more meaningful names to time series variables. You can also use LABEL statements to associate descriptive labels with your series variables.

For example, the series names for effective exchange rate (F_X_AM) and exchange rate index (F_X_AH) used by IMF can be given more descriptive names and labels by the following statements:

```
filename ifsfile 'host-specific-file-name' <host-options>;
proc datasource filetype=imfifsp infile=ifsfile interval=month
                out=exchange outcont=exchvars;
   where country in ('112','146','158') and partner=' ';
   keep  f_x_ah f_x_am;
   range from '01jun87'd to '01feb88'd;
   rename  f_x_ah=exrindex  f_x_am=effexr;
   label   f_x_ah='F_X_AH: Exchange Rate Index 1985=100'
           f_x_am='F_X_AM: Effective Exchange Rate(MERM)';
run;

title1 'Printout of OUTCONT= Showing New NAMEs and LABELs';
proc print data=exchvars noobs;
   var  name label length;
run;

title1 'Contents of OUT= Showing New NAMEs and LABELs';
proc contents data=exchange;
run;
```

The RENAME statement allows input names to be quoted strings. If the name of a series in the input file contains blanks or special characters that are not valid SAS

name syntax, use the SAS option VALIDVARNAME= ANY or put the series name in quotes to rename it. See the FAME example using rename in the "Selecting Time Series Variables – The KEEP and DROP Statements" section (page 452).

```
            Printout of OUTCONT= Showing New NAMEs and LABELs

          name                        label                      length

        EFFEXR      F_X_AM: Effective Exchange Rate(MERM)        5
        EXRINDEX    F_X_AH: Exchange Rate Index 1985=100         5
```

```
               Contents of OUT= Showing New NAMEs and LABELs

                           The CONTENTS Procedure

Data Set Name: WORK.EXCHANGE                          Observations:         27
Member Type:   DATA                                   Variables:            7
Engine:        V7                                     Indexes:              0
Created:       22:11 Saturday, May 30, 1998           Observation Length:   24
Last Modified: 22:11 Saturday, May 30, 1998           Deleted Observations: 0
Protection:                                           Compressed:           NO
Data Set Type:                                        Sorted:               NO
Label:


                -----Engine/Host Dependent Information-----

   Data Set Page Size:         8192
   Number of Data Set Pages:   1
   First Data Page:            1
   Max Obs per Page:           338
   Obs in First Data Page:     27
   Number of Data Set Repairs: 0
   File Name:                  /tmp/SAS_work2C5200004EF6/exchange.sas7bdat
   Release Created:            7.00.00P
   Host Created:               HP-UX
   Inode Number:               9622
   Access Permission:          rw-r--r--
   Owner Name:                 sasknh
   File Size (bytes):          16384


           -----Alphabetic List of Variables and Attributes-----

 #  Variable  Type  Len  Pos  Format   Label
 ------------------------------------------------------------------------
 3  country   Char    3    4           COUNTRY CODE
 4  csc       Char    1    7           CONTROL SOURCE CODE
 7  date      Num     4    0  MONYY7.  Date of Observation
 2  effexr    Num     5   17           F_X_AM: Effective Exchange Rate(MERM)
 1  exrindex  Num     5   12           F_X_AH: Exchange Rate Index 1985=100
 5  partner   Char    3    8           PARTNER COUNTRY CODE
 6  version   Char    1   11           VERSION CODE
```

Figure 10.6. Renaming and Labeling Variables

Notice that even though you changed the names of F_X_AH and F_X_AM to EXRINDEX and EFFEXR, respectively, you still used their old names in the KEEP and LABEL statements because renaming takes place at the output stage.

Changing the Lengths of Numeric Variables

The length attribute indicates the number of bytes the SAS System uses for storing the values of variables in output data sets. Therefore, the shorter the variable lengths, the more efficient the disk-space usage. However, there is a trade-off. The lengths of numeric variables are closely tied to their precision, and reducing their lengths arbitrarily can cause precision loss.

The DATASOURCE procedure uses default lengths for series variables appropriate to each file type. For example, the default lengths for numeric variables are 5 for IMFIFSP type files. In some cases, however, you may want to assign different lengths. Assigning lengths less than the defaults reduces memory and disk-space usage at the expense of reduced precision. Specifying lengths longer than the defaults increases the precision but causes the DATASOURCE procedure to use more memory and disk space. The following statements define a default length of 4 for all numeric variables in the IFSFILE and then assign a length of 6 to the exchange rate index:

```
filename ifsfile 'host-specific-file-name' <host-options>;
proc datasource filetype=imfifsp infile=ifsfile interval=month
                out=exchange outcont=exchvars;
   where country in ('112','146','158') and partner='   ';
   keep  f_x_am f_x_ah;
   range from '01jun87'd to '01feb88'd;
   rename  f_x_ah=exrindex  f_x_am=effexr;
   label   f_x_ah='F_X_AH: Exchange Rate Index 1985=100'
           f_x_am='F_X_AM: Effective Exchange Rate(MERM)';
   length _numeric_ 4; length f_x_ah 6;
run;

title1 'Printout of OUTCONT= Showing LENGTH Variable';
proc print data=exchvars noobs;
   var  name label length;
run;

title1 'Contents of the OUT= Data Set Showing LENGTHs';
proc contents data=exchange;
run;
```

```
        Printout of OUTCONT= Showing LENGTH Variable

  name                      label                    length

EFFEXR      F_X_AM: Effective Exchange Rate(MERM)       4
EXRINDEX    F_X_AH: Exchange Rate Index 1985=100        6
```

```
                Contents of the OUT= Data Set Showing LENGTHs

                            The CONTENTS Procedure

Data Set Name: WORK.EXCHANGE                          Observations:         27
Member Type:   DATA                                   Variables:            7
Engine:        V7                                     Indexes:              0
Created:       22:11 Saturday, May 30, 1998           Observation Length:   24
Last Modified: 22:11 Saturday, May 30, 1998           Deleted Observations: 0
Protection:                                           Compressed:           NO
Data Set Type:                                        Sorted:               NO
Label:


                 -----Engine/Host Dependent Information-----

   Data Set Page Size:         8192
   Number of Data Set Pages:   1
   First Data Page:            1
   Max Obs per Page:           338
   Obs in First Data Page:     27
   Number of Data Set Repairs: 0
   File Name:                  /tmp/SAS_work2C5200004EF6/exchange.sas7bdat
   Release Created:            7.00.00P
   Host Created:               HP-UX
   Inode Number:               9628
   Access Permission:          rw-r--r--
   Owner Name:                 sasknh
   File Size (bytes):          16384


            -----Alphabetic List of Variables and Attributes-----

 #  Variable  Type  Len  Pos  Format   Label
 ------------------------------------------------------------------------
 3  country   Char    3    8           COUNTRY CODE
 4  csc       Char    1   11           CONTROL SOURCE CODE
 7  date      Num     4    4  MONYY7.   Date of Observation
 2  effexr    Num     4    0           F_X_AM: Effective Exchange Rate(MERM)
 1  exrindex  Num     6   16           F_X_AH: Exchange Rate Index 1985=100
 5  partner   Char    3   12           PARTNER COUNTRY CODE
 6  version   Char    1   15           VERSION CODE
```

Figure 10.7. Changing the Lengths of Numeric Variables

The default lengths of the character variables are set to the minimum number of characters that can hold the longest possible value.

Syntax

The DATASOURCE procedure uses the following statements:

PROC DATASOURCE *options*;
KEEP *variable-list*;
*** DROP** *variable-list*;
KEEPEVENT *event-list*;
DROPEVENT *event-list*;
WHERE *where-expression*;
RANGE FROM *from* **TO** *to*;
ATTRIBUTE *variable-list attribute-list* . . . ;
FORMAT *variable-list format* . . . ;
LABEL *variable="label"* . . . ;
LENGTH *variable-list length* . . . ;
RENAME *old-name=new-name* . . . ;

The PROC DATASOURCE statement is required. All the rest of the statements are optional.

The DATASOURCE procedure uses two kinds of statements:

1. subsetting statements, which control what time series, time periods, and cross sections are extracted from the input data file
2. attribute statements, which control the attributes of the variables in the output SAS data set

The subsetting statements are the KEEP, DROP, KEEPEVENT, and DROPEVENT statements (which select output variables); the RANGE statement (which selects time ranges); and the WHERE statement (which selects cross sections). The attribute statements are the ATTRIBUTE, FORMAT, LABEL, LENGTH, and RENAME statements.

The statements and options used by PROC DATASOURCE are summarized in Table 10.2.

Table 10.2. Summary of Syntax

Description	Statement	Option
Input Data File Options		
specify the character set of the incoming data	PROC DATASOURCE	ASCII
	PROC DATASOURCE	EBCDIC

Description	Statement	Option
specify the type of input data file to read	PROC DATASOURCE	FILETYPE=
specify the fileref(s) of the input data file(s)	PROC DATASOURCE	INFILE=
specify the lrecl(s) of the input data files(s)	PROC DATASOURCE	LRECL=
specify the recfm(s) of the input data files(s)	PROC DATASOURCE	RECFM=
Output Data Set Options		
write the extracted time series data	PROC DATASOURCE	OUT=
output the descriptive information on the time series variables and cross sections	PROC DATASOURCE	OUTALL=
output the descriptive information on the cross sections	PROC DATASOURCE	OUTBY=
output the descriptive information on the time series variables	PROC DATASOURCE	OUTCONT=
write event-oriented data	PROC DATASOURCE	OUTEVENT=
control whether all or only selected series and cross sections be reported	PROC DATASOURCE	OUTSELECT=
create single indexes from BY variables for the OUT= data set	PROC DATASOURCE	INDEX
control the alignment of SAS Date values	PROC DATASOURCE	ALIGN=
Subsetting		
specify the periodicity of series to be extracted	PROC DATASOURCE	INTERVAL=
specify the time series variables to be included in the OUT= data set	KEEP	
specify the time series variables to be excluded from the OUT= data set	DROP	
specify the events to be included in the OUTEVENT= data set	KEEPEVENT	
specify the events to be excluded from the OUTEVENT= data set	DROPEVENT	
select cross sections for output	WHERE	
specify the time range of observations to be output	RANGE	

Description	Statement	Option
Assigning Attributes		
assign formats to the output variables	FORMAT	
	ATTRIBUTE	FORMAT=
assign labels to variables in the output data sets	LABEL	
	ATTRIBUTE	LABEL=
control the lengths of the output variables	LENGTH	
	ATTRIBUTE	LENGTH=
assign new names to the output variables	RENAME	

PROC DATASOURCE Statement

PROC DATASOURCE *options;*

The following options can be used in the PROC DATASOURCE statement:

ALIGN= *option*
controls the alignment of SAS dates used to identify output observations. The ALIGN= option allows the following values: BEGINNING|BEG|B, MIDDLE|MID|M, and ENDING|END|E. BEGINNING is the default.

ASCII
specifies the incoming data is ascii. This option is needed when the native character set of your host machine is ebcdic.

DBNAME= '*database name***'**
specifies the FAME database to access. Only use this option with the filetype=FAME option. The character string you specify on the DBNAME= option is passed through to FAME. Specify the value of this option as you would in accessing the database from within FAME software.

EBCDIC
specifies the incoming data is ebcdic. This option is needed when the native character set of your host machine is ascii.

FAMEPRINT
prints the FAME command file generated by PROC DATASOURCE and the log file produced by the FAME component of the interface system. Only use this option with the filetype=FAME option.

FILETYPE= *entry*
DBTYPE= *dbtype*
specifies the kind of input data file to process. See the "Supported File Types" section on page 482 for a list of supported file types. The FILETYPE= option is required.

INDEX

creates a set of single indexes from BY variables for the OUT= data set. Under some circumstances, creating indexes for a SAS data set may increase the efficiency in locating observations when BY or WHERE statements are used in subsequent steps. Refer to *SAS Language: Reference, Version 7, First Edition* for more information on SAS indexes. The INDEX option is ignored when no OUT= data set is created or when the data file does not contain any BY variables. The INDEX= data set option can be used to override the index variable definitions.

INFILE= *fileref*
INFILE= *(fileref1 fileref2 ... filerefn)*

specifies the *fileref* assigned to the input data file. The default value is DATAFILE. The fileref used in INFILE= option (or if no INFILE= option is specified, the fileref DATAFILE) must be associated with the physical data file in a FILENAME statement. (On some operating systems, the fileref assignment can be made with the system's control language, and a FILENAME statement may not be needed. Refer to *SAS Language: Reference Version 7, First Edition* for more details on the FILENAME statement). Physical data files can reside on tapes, disks, diskettes, CD-ROM, or other media.

For some file types, the data are distributed over several files. In this case, the INFILE= option is required, and it lists in parentheses the filerefs for each of the files making up the database. The order in which these FILEREFS are listed is important and must conform to the specifics of each file type as explained in the "Supported File Types" section on page 482.

LRECL= *lrecl*
LRECL= *(lrecl1 lrecl2 ... lrecln)*

The logical record length in bytes of the infile. Only use this if you need to override the default LRECL of the file. For some file types, the data are distributed over several files. In this case, the LRECL= option lists in parentheses the LRECLS for each of the files making up the database. The order in which these lrecls are listed is important and must conform to the specifics of each file type as explained in the "Supported File Types" section on page 482.

RECFM= *recfm*
RECFM= *(recfm1 recfm2 ... recfmn)*

The record format of the infile. Only use this if you need to override the default record format of the file. For some file types, the data are distributed over several files. In this case, the RECFM= option lists in parentheses the recfms for each of the files making up the database. The order in which these RECFMS are listed is important and must conform to the specifics of each file type as explained in the "Supported File Types" section on page 482. The possible values of RECFM are:

- F or FIXED for fixed length records
- N or BIN for binary records
- D or VAR for varying length records
- U or DEF for host default record format

- DOM_V or DOMAIN_VAR or BIN_V or BIN_VAR for unix binary record format

INTERVAL= *interval*
FREQUENCY= *interval*
TYPE= *interval*

specifies the periodicity of series selected for output to the OUT= data set. The OUT= data set created by PROC DATASOURCE can contain only time series with the same periodicity. Some data files contain time series with different periodicities; for example, a file may contain both monthly series and quarterly series. Use the INTERVAL= option to indicate which periodicity you want. If you want to extract series with different periodicities, use different PROC DATASOURCE invocations with the desired INTERVAL= options.

Common values for INTERVAL= are YEAR, QUARTER, MONTH, WEEK, and DAY. The values allowed, as well as the default value of the INTERVAL= option, depend on the file type. See the "Supported File Types" section on page 482 for the INTERVAL= values appropriate to the data file type you are reading.

OUT= *SAS-data-set*

names the output data set for the time series extracted from the data file. If none of the output data set options are specified, including the OUT= data set itself, an OUT= data set is created and named according to the DATA*n* convention. However, when you create any of the other output data sets, such as OUTCONT=, OUTBY=, OUTALL=, or OUTEVENT=, you must explicitly specify the OUT= data set; otherwise, it will not be created. See the "OUT= Data Set" section on page 476 for further details.

OUTALL= *SAS-data-set*

writes information on the contents of the input data file to an output data set. The OUTALL= data set includes descriptive information, time ranges, and observation counts for all the time series within each BY group. By default, no OUTALL= data set is created.

The OUTALL= data set contains the Cartesian product of the information output by the OUTCONT= and OUTBY= options. In data files for which there are no cross sections, the OUTALL= and OUTCONT= data sets are almost equivalent, except that OUTALL= data set also reports time ranges and observation counts of series. See the "OUTALL= Data Set" section on page 479 for further details.

OUTBY= *SAS-data-set*

writes information on the BY variables to an output data set. The OUTBY= data set contains the list of cross sections in the database delimited by the unique set of values that the BY variables assume. Unless the OUTSELECT=OFF option is present, only the selected BY groups get written to the OUTBY= data set. If you omit the OUTBY= option, no OUTBY= data set is created. See the "OUTBY= Data Set" section on page 478 for further details.

OUTCONT= *SAS-data-set*

writes information on the contents of the input data file to an output data set. By de-

fault, the OUTCONT= data set includes descriptive information on all of the unique series of the selected periodicity in the data file. When the OUTSELECT=OFF option is omitted, the OUTCONT= data set includes observations only for the series selected for output to the OUT= data set. By default, no OUTCONT= data set is created. See the "OUTCONT= Data Set" section on page 477 for further details.

OUTEVENT= *SAS-data-set*

names the output data set to output event-oriented time series data. This option can only be used when CRSP stock files are being processed. For all other file types, it will be ignored. See the "OUTEVENT= Data Set" section on page 480 for further details.

OUTSELECT= ON | OFF

determines whether to output all observations (OUTSELECT=OFF) or only those corresponding to the selected time series and selected BY groups (OUTSELECT=ON) to OUTCONT=, OUTBY=, and OUTALL= data sets. The default is OUTSELECT=ON. The OUTSELECT= option is only relevant when any one of the auxiliary data sets is specified. The option writes observations to OUTCONT=, OUTBY=, and OUTALL= data sets for only the selected time series and selected BY groups if it is set ON. The OUTSELECT= option is only relevant when any one of the OUTCONT=, OUTBY= and OUTALL= options are specified. The default is OUTSELECT=ON.

KEEP Statement

KEEP *variable-list;*

The KEEP statement specifies which variables in the data file are to be included in the OUT= data set. Only the time series and event variables can be specified in a KEEP statement. All the BY variables and the time ID variable DATE are always included in the OUT= data set; they cannot be referenced in a KEEP statement. If they are referenced, a warning message is given and the reference is ignored.

The variable list can contain variable names or name range specifications. See the section "Variable Lists" on page 475 for details.

There is a default KEEP list for each file type. Usually, descriptor type variables, like footnotes, are not included in the default KEEP list. If you give a KEEP statement, the default list becomes undefined.

Only one KEEP or one DROP statement can be used. KEEP and DROP are mutually exclusive.

You can also use the KEEP= data set option to control which variables to include in the OUT= data set. However, the KEEP statement differs from the KEEP= data set option in several aspects:

- The KEEP statement selection is applied before variables are read from the data file, while the KEEP= data set option selection is applied after variables

are read and as they are written to the OUT= data set. Therefore, using the KEEP statement instead of the KEEP= data set option is much more efficient.

- If the KEEP statement causes no series variables to be selected, then no observations are output to the OUT= data set.
- The KEEP statement variable specifications are applied to each cross section independently. This behavior may produce different variables than those produced by the KEEP= data set option when order-range variable list specifications are used.

DROP Statement

DROP *variable-list;*

The DROP statement specifies that some variables be excluded from the OUT= data set. Only the time series and event variables can be specified in a DROP statement. None of the BY variables or the time ID variable DATE can be excluded from the OUT= data set. If they are referenced in a DROP statement, a warning message is given and the reference is ignored. Use the WHERE statement for selection based on BY variables, and use the RANGE statement for date selections.

The variable list can contain variable names or name range specifications. See the section “Variable Lists” on page 475 for details.

Only one DROP or one KEEP statements can be used. KEEP and DROP are mutually exclusive.

There is a default KEEP list for each file type. Usually, descriptor type variables, like footnotes, are not included in the default KEEP list. If you specify a DROP statement, the default list becomes undefined.

You can also use the DROP= data set option to control which variables to exclude from the OUT= data set. However, the DROP statement differs from the DROP= data set option in several aspects:

- The DROP statement selection is applied before variables are read from the data file, while the DROP= data set option selection is applied after variables are read and as they are written to the OUT= data set. Therefore, using the DROP statement instead of the DROP= data set option is much more efficient.
- If the DROP statement causes all series variables to be excluded, then no observations are output to the OUT= data set.
- The DROP statement variable specifications are applied to each cross section independently. This behavior may produce different variables than those produced by the DROP= data set option when order-range variable list specifications are used.

KEEPEVENT Statement

KEEPEVENT *variable-list;*

The KEEPEVENT statement specifies which event variables in the data file are to be included in the OUTEVENT= data set. As a result, the KEEPEVENT statement is valid only for data files containing event-oriented time series data, that is, only for CRSP files. All the BY variables, the time ID variable DATE and the event-grouping variable EVENT are always included in the OUTEVENT= data set. These variables can not be referenced in the KEEPEVENT statement. If any of these variables are referenced, a warning message is given and the reference is ignored.

The variable list can contain variable names or name range specifications. See the section “Variable Lists” on page 475 for details.

Only one KEEPEVENT or one DROPEVENT statement can be used. KEEPEVENT and DROPEVENT are mutually exclusive.

You can also use the KEEP= data set option to control which event variables to include in the OUTEVENT= data set. However, the KEEPEVENT statement differs from the KEEP= data set option in several aspects:

- The KEEPEVENT statement selection is applied before variables are read from the data file, while the KEEP= data set option selection is applied after variables are read and as they are written to the OUTEVENT= data set. Therefore, using the KEEPEVENT statement instead of the KEEP= data set option is much more efficient.
- If the KEEPEVENT statement causes no event variables to be selected, then no observations are output to the OUTEVENT= data set.

DROPEVENT Statement

DROPEVENT *variable-list;*

The DROPEVENT statement specifies that some event variables be excluded from the OUTEVENT= data set. As a result, the DROPEVENT statement is valid only for data files containing event-oriented time series data, that is, only for CRSP files. All the BY variables, the time ID variable DATE, and the event-grouping variable EVENT are always included in the OUTEVENT= data set. These variables cannot be referenced in the DROPEVENT statement. If any of these variables are referenced, a warning message is given and the reference is ignored.

The variable list can contain variable names or name range specifications. See the section “Variable Lists” on page 475 for details.

Only one DROPEVENT or one KEEPEVENT statement can be used. DROPEVENT and KEEPEVENT are mutually exclusive.

You can also use the DROP= data set option to control which event variables to exclude from the OUTEVENT= data set. However, the DROPEVENT statement differs from the DROP= data set option in several aspects:

- The DROPEVENT statement selection is applied before variables are read from the data file, while the DROP= data set option selection is applied after variables are read and as they are written to the OUTEVENT= data set. Therefore, using the DROPEVENT statement instead of the DROP= data set option is much more efficient.
- If the DROPEVENT statement causes all series variables to be excluded, then no observations are output to the OUTEVENT= data set.

WHERE Statement

WHERE *where-expression;*

The WHERE statement specifies conditions that BY variables must satisfy in order for a cross section to be included in the OUT= and OUTEVENT= data sets. By default, all BY groups are selected.

The *where-expression* must refer only to BY variables defined for the file type you are reading. The section "Supported File Types" on page 482 lists the names of the BY variables for each file type.

For example, DOTS (Direction of Trade Statistics) files, distributed by International Monetary Fund, have four BY variables: COUNTRY, CSC, PARTNER, and VERSION. Both COUNTRY and PARTNER are three-digit country codes. To select the direction of trade statistics of the United States (COUNTRY='111') with Turkey (COUNTRY='186'), Japan (COUNTRY='158'), and the oil exporting countries group (COUNTRY='985'), you should specify

```
where country='111' and partner in ('186','158','985');
```

You can use any SAS language operators and special WHERE expression operators in the WHERE statement condition. Refer to *SAS Language: Reference, Version 7, First Edition* for a more detailed discussion of WHERE expressions.

If you want to see the names of the BY variables and the values they assume for each cross section, you can first run PROC DATASOURCE with only the OUTBY= option. The information contained in the OUTBY= data set will aid you in selecting the appropriate BY groups for subsequent PROC DATASOURCE steps.

RANGE Statement

RANGE FROM *from* **TO** *to;*

The RANGE statement selects the time range of observations written to the OUT= and OUTEVENT= data sets. The *from* and *to* values can be SAS date, time, or datetime constants, or they can be specified as *year* or *year***:***period*, where *year* is a two-digit or four-digit year, and *period* (when specified) is a period within the year corresponding to the INTERVAL= option. (For example, if INTERVAL=QTR, then *period* refers to quarters.) When *period* is omitted, the beginning of the year is assumed for the *from* value, and the end of the year is assumed for the *to* value.

If a 2 digit year is specified, PROC DATASOURCE complies to year 2000 guidelines by using the current value of the YEARCUTOFF option to determine the century of your data. Warnings are issued in the SAS log whenever DATASOURCE needs to determine the century from a 2 digit year specification.

The default YEARCUTOFF is 1920. To use a different yearcutoff, specify

```
options yearcutoff=yyyy;
```

where yyyy is the yearcutoff you want to use. See the *SAS Language: Reference, Version 7, First Edition* for a more detailed discussion of the YEARCUTOFF option.

Both the FROM and TO specifications are optional, and both the FROM and TO keywords are optional. If the FROM limit is omitted, the output observations start with the minimum date for which data is available for any selected series. Similarly, if the TO limit is omitted, the output observations end with the maximum date for which data are available.

The following are some examples of RANGE statements:

```
range from 1980 to 1990;
range 1980 - 1990;
range from 1980;
range 1980;
range to 1990;
range to 1990:2;
range from '31aug89'd to '28feb1990'd;
```

The RANGE statement applies to each BY group independently. If all the selected series contain no data in the specified range for a given BY group, then there will be no observations for that BY group in the OUT= and OUTEVENT= data sets.

If you want to know the time ranges for which periodic time series data is available, you can first run PROC DATASOURCE with the OUTBY= or OUTALL= options. OUTBY= data set reports the union of the time ranges over all the series within each BY group, while the OUTALL= data set gives time ranges for each series separately in each BY group.

ATTRIBUTE Statement

ATTRIBUTE *variable-list attribute-list ... ;*

The ATTRIBUTE statement assigns formats, labels, and lengths to variables in the output data sets.

The *variable-list* can contain variable names and variable name range specifications. See the section "Variable Lists" on page 475 for details. The attributes specified in the following attribute list apply to all variables in the variable list:

An *attribute-list* consists of one or more of the following options:

FORMAT= *format*
associates a format with variables in *variable-list*. The *format* can be either a standard SAS format or a format defined with the FORMAT procedure. The default formats for variables depend on the file type.

LABEL= *"label"*
assigns a label to the variables in the variable list. The default labels for variables depend on the file type. Labels can be up to 256 bytes in length.

LENGTH= *length*
specifies the number of bytes used to store the values of variables in the variable list. The default lengths for numeric variables depend on the file type. Usually default lengths are set to 5 bytes. (For CRSP files, the default lengths are 6 bytes).

The length specification also controls the amount of memory that PROC DATASOURCE uses to hold variable values while processing the input data file. Thus, specifying a LENGTH= value smaller than the default will reduce both the disk space taken up by the output data sets and the amount of memory used by the PROC DATASOURCE step, at the cost of reduced precision of output data values.

FORMAT Statement

FORMAT *variable-list format ... ;*

The FORMAT statement assigns formats to variables in output data sets. The *variable-list* can contain variable names and variable name range specifications. See the section "Variable Lists" on page 475 for details. The format specified applies to all variables in the variable list.

A single FORMAT statement can assign the same format to several variables or different formats to different variables. The FORMAT statement can use standard SAS formats or formats defined using the FORMAT procedure.

Any later format specification for a variable, using either the FORMAT statement or the FORMAT= option in the ATTRIBUTE statement, always overrides the previous one.

LABEL Statement

LABEL *variable = "label" ... ;*

The LABEL statement assigns SAS variable labels to variables in the output data sets. You can give labels for any number of variables in a single LABEL statement. The default labels for variables depend on the file type. Extra long labels (> 256 bytes) reside in the OUTCONT data set as the DESCRIPT variable.

Any later label specification for a variable, using either the LABEL statement or the LABEL= option in the ATTRIBUTE statement, always overrides the previous one.

LENGTH Statement

LENGTH *variable-list length ... ;*

The LENGTH statement, like the LENGTH= option in the ATTRIBUTE statement, specifies the number of bytes used to store values of variables in output data sets. The default lengths for numeric variables depend on the file type. Usually default lengths are set to 5 bytes. (For CRSP files, the default lengths are 6 bytes).

The default lengths of character variables are defined as the minimum number of characters that can hold the longest possible value.

For some file types, the LENGTH statement also controls the amount of memory used to store values of numeric variables while processing the input data file. Thus, specifying LENGTH values smaller than the default will reduce both the disk space taken up by the output data sets and the amount of memory used by the PROC DATASOURCE step, at the cost of reduced precision of output data values.

Any later length specification for a variable, using either the LENGTH statement or the LENGTH= option in the ATTRIBUTE statement, always overrides the previous one.

RENAME Statement

RENAME *old-name = new-name ... ;*

The RENAME statement is used to change the names of variables in the output data sets. Any number of variables can be renamed in a single RENAME statement. The most recent RENAME specification overrides any previous ones for a given variable. The new-name is limited to thirty-two characters.

Renaming of variables is done at the output stage. Therefore, you need to use the old variable names in all other PROC DATASOURCE statements. For example, the series variable names DATA1-DATA350 used with annual COMPUSTAT files are not very descriptive, so you may choose to rename them to reflect the financial aspect

they represent. You may rename "DATA51" to "INVESTTAX" with the RENAME statement

```
rename data51=investtax;
```

since it contains investment tax credit data. However, in all other DATASOURCE statements, you must use the old name, DATA51.

Details

Variable Lists

Variable lists used in PROC DATASOURCE statements can consist of any combination of variable names and name range specifications. Items in variable lists can have the following forms:

- a name, for example, PZU.
- an alphabetic range *name1-name2*. For example, A-DZZZZZZZ specifies all variables with names starting with A, B, C, or D.
- a prefix range *prefix*:. For example, IP: selects all variables with names starting with the letters IP.
- an order range *name1--name2*. For example, GLR72--GLRD72 specifies all variables in the input data file between GLR72 and GRLD72 inclusive.
- a numeric order range *name1*-NUMERIC-*name2*. For example, GLR72-NUMERIC-GLRD72 specifies all numeric variables between GLR72 and GRLD72 inclusive.
- a character order range *name1*-CHARACTER-*name2*. For example, GLR72-CHARACTER-GLRD72 specifies all character variables between GLR72 and GRLD72 inclusive.
- one of the keywords _NUMERIC_, _CHARACTER_, or _ALL_. _NUMERIC_ specifies all numeric variables. _CHARACTER_ specifies all character variables. _ALL_ specifies all variables.

To determine the order of series in a data file, run PROC DATASOURCE with the OUTCONT= option, and print the output data set. Note that order and alphabetic range specifications are inclusive, meaning that the beginning and ending names of the range are also included in the variable list.

For order ranges, the names used to define the range must actually name variables in the input data file. For alphabetic ranges, however, the names used to define the range need not be present in the data file.

Note that variable specifications are applied to each cross section independently. This may cause the order-range variable list specification to behave differently than its DATA step and data set option counterparts. This is because PROC DATASOURCE knows which variables are defined for which cross sections, while the DATA step applies order range specification to the whole collection of time series variables.

If the ending variable name in an order range specification is not in the current cross section, all variables starting from the beginning variable to the last variable defined in that cross section get selected. If the first variable is not in the current cross section, then order range specification has no effect for that cross section.

The variable names used in variable list specifications can refer to either series names appearing in the input data file or to the SAS names assigned to series data fields

internally if the series names are not recorded to the INFILE= file. When the latter is the case, internally defined variable names are listed in the section "Data Files" later in this chapter.

The following are examples of the use of variable lists:

```
keep  ip: pw112-pw117 pzu;
drop  data1-data99  data151-data350;
length data1-numeric-aftnt350  ucode 4;
```

The first statement keeps all the variables starting with IP:, all the variables between PW112 and PW117 including the PW112 and PW117 themselves, and a single variable PZU. The second statement drops all the variables that fall alphabetically between DATA1 and DATA99, and DATA151 and DATA350. Finally, the third statement assigns a length of 4 bytes to all the numeric variables defined between DATA1 and AFTNT350, and UCODE.

OUT= Data Set

The OUT= data set can contain the following variables:

- the BY variables, which identify cross-sectional dimensions when the input data file contains time series replicated for different values of the BY variables. Use the BY variables in a WHERE statement to process the OUT= data set by cross sections. The order in which BY variables are defined in the OUT= data set corresponds to the order in which the data file is sorted.
- DATE, a SAS date-, time-, or datetime- valued variable that reports the time period of each observation. The values of the DATE variable may span different time ranges for different BY groups. The format of the DATE variable depends on the INTERVAL= option.
- the periodic time series variables, which are included in the OUT= data set only if they have data in at least one selected BY group and they are not discarded by a KEEP or DROP statement
- the event variables, which are included in the OUT= data set if they are not discarded by a KEEP or DROP statement. By default, these variables are not output to OUT= data set.

The values of BY variables remain constant in each cross section. Observations within each BY group correspond to the sampling of the series variables at the time periods indicated by the DATE variable.

You can create a set of single indexes for the OUT= data set by using the INDEX option, provided there are BY variables. Under some circumstances, this may increase the efficiency of subsequent PROC and DATA steps that use BY and WHERE statements. However, there is a cost associated with creation and maintenance of indexes. The *SAS Language: Reference, Version 7, First Edition* lists the conditions under which the benefits of indexes outweigh the cost.

With data files containing cross sections, there can be various degrees of overlap among the series variables. One extreme is when all the series variables contain data for all the cross sections. In this case, the output data set is very compact. In the other extreme case, however, the set of time series variables are unique for each cross section, making the output data set very sparse, as depicted in Figure 10.8.

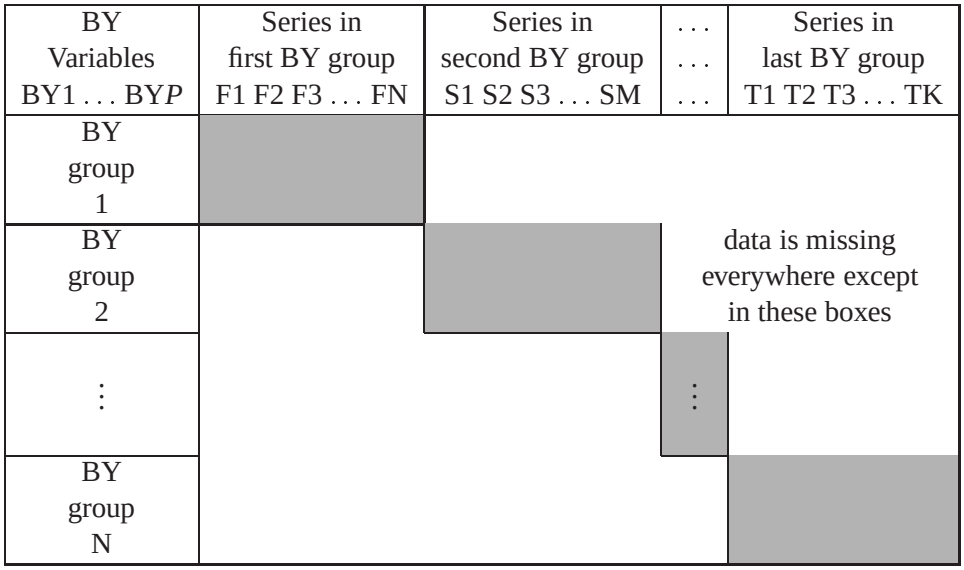


Figure 10.8. The OUT= Data Set containing unique Series for each BY Group

The data in Figure 10.8 can be represented more compactly if cross-sectional information is incorporated into series variable names.

OUTCONT= Data Set

The OUTCONT= data set contains descriptive information for the time series variables. This descriptive information includes various attributes of the time series variables. The OUTCONT= data set contains the following variables:

- NAME, a character variable that contains the series name.
- KEPT, a numeric variable that indicates whether the series was selected for output by the DROP or KEEP statements, if any. KEPT will usually be the same as SELECTED, but may differ if a WHERE statement is used.
- SELECTED, a numeric variable that indicates whether the series is selected for output to the OUT= data set. The series is included in the OUT= data set (SELECTED=1) if it is kept (KEPT=1) and it has data for at least one selected BY group.
- TYPE, a numeric variable that indicates the type of the time series variable. TYPE=1 for numeric series; TYPE=2 for character series.
- LENGTH, a numeric variable that gives the number of bytes allocated for the series variable in the OUT= data set.

- VARNUM, a numeric variable that gives the variable number of the series in the OUT= data set. If the series variable is not selected for output (SELECTED=0), then VARNUM has a missing value. Likewise, if no OUT= option is given, VARNUM has all missing values.
- LABEL, a character variable that contains the label of the series variable. LABEL contains only the first 256 characters of the labels. If they are longer than 256 characters, then the variable, DESCRIPT, is defined to hold the whole length of series labels. Note that if a data file assigns different labels to the same series variable within different cross sections, only the first occurrence of labels will be transferred to the LABEL column.
- the variables FORMAT, FORMATL, and FORMATD, which give the format name, length, and number of format decimals, respectively.
- the GENERIC variables, whose values may vary from one series to another, but whose values remain constant across BY groups for the same series.

By default, the OUTCONT= data set contains observations for only the selected series, that is, for series where SELECTED=1. If the OUTSELECT=OFF option is specified, the OUTCONT= data set contains one observation for each unique series of the specified periodicity contained in the input data file.

If you do not know what series are in the data file, you can run PROC DATASOURCE with the OUTCONT= option and OUTSELECT=OFF. The information contained in the OUTCONT= data set can then help you to determine which time series data you want to extract.

OUTBY= Data Set

The OUTBY= data set contains information on the cross sections contained in the input data file. These cross sections are represented as BY groups in the OUT= data set. The OUTBY= data set contains the following variables:

- the BY variables, whose values identify the different cross sections in the data file. The BY variables depend on the file type.
- BYSELECT, a numeric variable that reports the outcome of the WHERE statement condition for the BY variable values for this observation. The value of BYSELECT is 1 for BY groups selected by the WHERE statement for output to the OUT= data set and is 0 for BY groups that are excluded by the WHERE statement. BYSELECT is added to the data set only if a WHERE statement is given. When there is no WHERE statement, then all the BY groups are selected.
- ST_DATE, a numeric variable that gives the starting date for the BY group. The starting date is the earliest of the starting dates of all the series that have data for the current BY group.
- END_DATE, a numeric variable that gives the ending date for the BY group. The ending date is the latest of the ending dates of all the series that have data for the BY group.

- NTIME, a numeric variable that gives the number of time periods between ST_DATE and END_DATE, inclusive. Usually, this is the same as NOBS, but they may differ when time periods are not equally spaced and when the OUT= data set is not specified. NTIME is a maximum limit on NOBS.
- NOBS, a numeric variable that gives the number of time series observations in OUT= data set between ST_DATE and END_DATE, inclusive. When a given BY group is discarded by a WHERE statement, the NOBS variable corresponding to this BY group becomes 0, since the OUT= data set does not contain any observations for this BY group. Note that BYSELECT=0 for every discarded BY group.
- NINRANGE, a numeric variable that gives the number of observations in the range (*from*,*to*) defined by the RANGE statement. This variable is only added to the OUTBY= data set when the RANGE statement is specified.
- NSERIES, a numeric variable that gives the total number of unique time series variables having data for the BY group.
- NSELECT, a numeric variable that gives the total number of selected time series variables having data for the BY group.
- the generic variables, whose values remain constant for all the series in the current BY group.

In this list, you can only control the attributes of the BY and GENERIC variables.

The variables NOBS, NTIME, and NINRANGE give observation counts, while the variables NSERIES and NSELECT give series counts.

By default, observations for only the selected BY groups (where BYSELECT=1) are output to the OUTBY= data set, and the date and time range variables are computed over only the selected time series variables. If the OUTSELECT=OFF option is specified, the OUTBY= data set contains an observation for each BY group, and the date and time range variables are computed over all the time series variables.

For file types that have no BY variables, the OUTBY= data set contains one observation giving ST_DATE, END_DATE, NTIME, NOBS, NINRANGE, NSERIES, and NSELECT for all the series in the file.

If you do not know the BY variable names or their possible values, you can do an initial run of PROC DATASOURCE with the OUTBY= option. The information contained in the OUTBY= data set can help you design your WHERE expression and RANGE statement for the subsequent executions of PROC DATASOURCE to obtain different subsets of the same data file.

OUTALL= Data Set

The OUTALL= data set combines and expands the information provided by the OUTCONT= and OUTBY= data sets. That is, the OUTALL= data set not only reports the OUTCONT= information separately for each BY group, but also reports the OUTBY= information separately for each series. Each observation in the OUTBY= data set gets expanded to NSERIES or NSELECT observations in the OUTALL= data set, depending on whether the OUTSELECT=OFF option is specified.

By default, only the selected BY groups and series are included in the OUTALL= data set. If the OUTSELECT=OFF option is specified, then all the series within all the BY groups are reported.

The OUTALL= data set contains all the variables defined in the OUTBY= and OUTCONT= data sets and also contains the GENERIC variables (whose values may vary from one series to another and also from one BY group to another). Another additional variable is BLKNUM, which gives the data block number in the data file containing the series variable.

The OUTALL= data set is useful when BY groups do not contain the same time series variables or when the time ranges for series change across BY groups.

You should be careful in using the OUTALL= option, since the OUTALL= data set can get very large for many file types. Some file types have the same series and time ranges for each BY group; the OUTALL= option should not be used with these file types. For example, you should not specify the OUTALL= option with COMPUSTAT files, since all the BY groups contain the same series variables.

The OUTALL= and OUTCONT= data sets are equivalent when there are no BY variables, except that the OUTALL= data set contains extra information about the time ranges and observation counts of the series variables.

OUTEVENT= Data Set

The OUTEVENT= data set is used to output event-oriented time series data. Events occurring at discrete points in time are recorded along with the date they occurred. Only CRSP stock files contain event-oriented time series data. For all other types of files, the OUTEVENT= option is ignored.

The OUTEVENT= data set contains the following variables:

- the BY variables, which identify cross-sectional dimensions when the input data file contains time series replicated for different values of the BY variables. Use the BY variables in a WHERE statement to process the OUTEVENT= data set by cross sections. The order in which BY variables are defined in the OUTEVENT= data set corresponds to the order in which the data file is sorted.
- DATE, a SAS date-, time- or datetime- valued variable that reports the discrete time periods at which events occurred. The format of the DATE variable depends on the INTERVAL= option, and should accurately report the date based on the SAS YEARCUTOFF option. The default value for YEARCUTOFF is 1920. The dates used may span up to 250 years.
- EVENT, a character variable that contains the event group name. The EVENT variable is another cross-sectional variable.
- the event variables, included in the OUTEVENT= data set only if they have data in at least one selected BY group, are not discarded by a KEEPEVENT or DROPEVENT statement.

Note that each event group contains a nonoverlapping set of event variables; therefore, the OUTEVENT= data set is very sparse. You should exercise care when selecting event variables to be included in the OUTEVENT= data set.

Also note that even though the OUTEVENT= data set can not contain any periodic time series variables, the OUT= data set can contain event variables if they are explicitly specified in a KEEP statement. In other words, you can specify event variables in a KEEP statement, but you cannot specify periodic time series variables in a KEEPEVENT statement.

While variable selection for OUT= and OUTEVENT= data sets are controlled by a different set of statements (KEEP versus KEEPEVENT or DROP versus DROPEVENT), cross-section and range selections are controlled by the same statements. In other words, the WHERE and the RANGE statements are effective for both output data sets.

Supported File Types

PROC DATASOURCE can process only certain kinds of data files. For certain time series databases, the DATASOURCE procedure has built-in information on the layout of files comprising the database. PROC DATASOURCE knows how to read only these kinds of data files. To access these databases, you must indicate the data file type in the FILETYPE= option. For more detailed information, see the corresponding document for each filetype. See the section “References” on page 535.

The currently supported file types are summarized in Table 10.3.

Table 10.3. Supported File Types

Supplier	FILETYPE=	Description
BEA	BEANIPA	National Income and Product Accounts Tape Format
	BEANIPAD	National Income and Product Accounts Diskette Format
BLS	BLSCPI	Consumer Price Index Surveys
	BLSWPI	Producer Price Index Survey
	BLSEENA	National Employment, Hours, and Earnings Survey
	BLSEESA	State and Area Employment Hours and Earnings Survey
DRI	DRIBASIC	Basic Economic (formerly CITIBASE) Data Files
	CITIBASE	Tape Format CITIBASE Data Files
	DRIDDS	DRI Data Delivery Service Time Series
	CITIDISK	PC Diskette format CITIBASE Databases
CRSP	CRSPDBS	CRSP Daily Binary Security File Format
	CRSPDBI	CRSP Daily Binary Calendar&Indices File Format
	CRSPDBA	CRSP Daily Binary File Annual Data Format
	CRSPMBS	CRSP Monthly Binary Security File Format
	CRSPMBI	CRSP Monthly Binary Calendar&Indices File Format
	CRSPMBA	CRSP Monthly Binary File Annual Data Format
CRSP	CRSPDCS	CRSP Daily Character Security File Format
	CRSPDCI	CRSP Daily Character Calendar&Indices File Format
	CRSPDCA	CRSP Daily Character File Annual Data Format
	CRSPMCS	CRSP Monthly Character Security File Format
	CRSPMCI	CRSP Monthly Character Calendar&Indices File Format
	CRSPMCA	CRSP Monthly Character File Annual Data Format
CRSP	CRSPDIS	CRSP Daily IBM Binary Security File Format
	CRSPDII	CRSP Daily IBM Binary Calendar&Indices File Format
	CRSPDIA	CRSP Daily IBM Binary File Annual Data Format
	CRSPMIS	CRSP Monthly IBM Binary Security File Format
	CRSPMII	CRSP Monthly IBM Binary Calendar&Indices File Format
	CRSPMIA	CRSP Monthly IBM Binary File Annual Data Format
CRSP	CRSPMVS	CRSP Monthly VAX Binary Security File Format
	CRSPMVI	CRSP Monthly VAX Binary Calendar&Indices File Format
	CRSPMVA	CRSP Monthly VAX Binary File Annual Data Format
	CRSPDVS	CRSP Daily VAX Binary Security File Format

Table 10.3. (continued)

Supplier	FILETYPE=	Description
	CRSPDVI	CRSP Daily VAX Binary Calendar&Indices File Format
	CRSPDVA	CRSP Daily VAX Binary File Annual Data Format
CRSP	CRSPMUS	CRSP Monthly UNIX Binary Security File Format
ACCESS97		CRSP ACCESS97 Monthly Security File Format
	CRSPMUI	CRSP Monthly UNIX Binary Calendar&Indices File Format
		CRSP ACCESS97 Monthly Calendar&Indices File Format
	CRSPMUA	CRSP Monthly UNIX Binary File Annual Data Format
		CRSP ACCESS97 Monthly Annual Data File Format
CRSP	CRSPDUS	CRSP Daily UNIX Binary Security File Format
ACCESS97		CRSP ACCESS97 Daily Security File Format
	CRSPDUI	CRSP Daily UNIX Binary Calendar&Indices File Format
		CRSP ACCESS97 Daily Calendar&Indices File Format
	CRSPDUA	CRSP Daily UNIX Binary File Annual Data Format
		CRSP ACCESS97 Daily Annual Data File Format
CRSP	CRSPMOS	CRSP Monthly Old Character Security File Format
	CRSPMOI	CRSP Monthly Old Character Calendar&Indices File Format
	CRSPMOA	CRSP Monthly Old Character File Annual Data Format
	CRSPDOS	CRSP Daily Old Character Security File Format
	CRSPDOI	CRSP Daily Old Character Calendar&Indices File Format
	CRSPDOA	CRSP Daily Old Character File Annual Data Format
CRSP	CR95MIS	CRSP 1995 Monthly IBM Binary Security File Format
	CR95MII	CRSP 1995 Monthly IBM Binary Calendar&Indices File Format
	CR95MIA	CRSP 1995 Monthly IBM Binary File Annual Data Format
	CR95DIS	CRSP 1995 Daily IBM Binary Security File Format
	CR95DII	CRSP 1995 Daily IBM Binary Calendar&Indices File Format
	CR95DIA	CRSP 1995 Daily IBM Binary File Annual Data Format
CRSP	CR95MVS	CRSP 1995 Monthly VAX Binary Security File Format
	CR95MVI	CRSP 1995 Monthly VAX Binary Calendar&Indices File Format
	CR95MVA	CRSP 1995 Monthly VAX Binary File Annual Data Format
	CR95DVS	CRSP 1995 Daily VAX Binary Security File Format
	CR95DVI	CRSP 1995 Daily VAX Binary Calendar&Indices File Format
	CR95DVA	CRSP 1995 Daily VAX Binary File Annual Data Format
CRSP	CR95MUS	CRSP 1995 Monthly UNIX Binary Security File Format
	CR95MUI	CRSP 1995 Monthly UNIX Binary Calendar&Indices File Format

Table 10.3. (continued)

Supplier	FILETYPE=	Description
	CR95MUA	CRSP 1995 Monthly UNIX Binary File Annual Data Format
	CR95DUS	CRSP 1995 Daily UNIX Binary Security File Format
	CR95DUI	CRSP 1995 Daily UNIX Binary Calendar&Indices File Format
	CR95DUA	CRSP 1995 Daily UNIX Binary File Annual Data Format
CRSP	CR95MSS	CRSP 1995 Monthly VMS Binary Security File Format
	CR95MSI	CRSP 1995 Monthly VMS Binary Calendar&Indices File Format
	CR95MSA	CRSP 1995 Monthly VMS Binary File Annual Data Format
	CR95DSS	CRSP 1995 Daily VMS Binary Security File Format
	CR95DSI	CRSP 1995 Daily VMS Binary Calendar&Indices File Format
	CR95DSA	CRSP 1995 Daily VMS Binary File Annual Data Format
CRSP	CR95MAS	CRSP 1995 Monthly ALPHA Binary Security File Format
	CR95MAI	CRSP 1995 Monthly ALPHA Binary Calendar&Indices File Format
	CR95MAA	CRSP 1995 Monthly ALPHA Binary File Annual Data Format
	CR95DAS	CRSP 1995 Daily ALPHA Binary Security File Format
	CR95DAI	CRSP 1995 Daily ALPHA Binary Calendar&Indices File Format
	CR95DAA	CRSP 1995 Daily ALPHA Binary File Annual Data Format
Haver	HAVER	Haver Analytics Data Files
IMF	IMFIFSP	International Financial Statistics, Packed Format
	IMFDOTSP	Direction of Trade Statistics, Packed Format
	IMFBOPSP	Balance of Payment Statistics, Packed Format
	IMFGFSP	Government Finance Statistics, Packed Format
OECD	OECDANA	OECD Annual National Accounts Tape Format
	OECDQNA	OECD Quarterly National Accounts Tape Format
	OECDMEI	OECD Main Economic Indicators Tape Format
S&P	CSAIBM	COMPUSTAT Annual, IBM 360&370 Format
	CS48QIBM	COMPUSTAT 48 Quarter, IBM 360&370 Format
	CSAUC	COMPUSTAT Annual, Universal Character Format
	CS48QUC	COMPUSTAT 48 Quarter, Universal Character Format

Data supplier abbreviations used in Table 10.3 are

Abbreviation	Supplier
BEA	Bureau of Economic Analysis, U.S. Department of Commerce
BLS	Bureau of Labor Statistics, U.S. Department of Labor
CRSP	Center for Research in Security Prices
DRI	DRIMcGraw-Hill
FAME	FAME Information Services, Inc
Haver	Haver Analytics Inc.
IMF	International Monetary Fund
OECD	Organization for Economic Cooperation and Development
S&P	Standard & Poor's Compustat Services Inc.

BEA Data Files

The Bureau of Economic Analysis, U.S. Department of Commerce, supplies national income, product accounting, and various other macro economic data at the regional, national, and international levels in the form of data files with various formats and on various media.

The following BEA data file types are supported:

FILETYPE=BEANIPA–National Income and Product Accounts Tape Format

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	
BY variables	PARTNO	Part Number of Publication, Integer Portion of the Table Number, 1-9 (character)
	TABNUM	Table Number Within Part, Decimal Portion of the Table Number, 1-24 (character)
Series Variables	Series variable names are constructed by concatenating table number suffix, line and column numbers within each table. An underscore (_) prefix is also added for readability.	

FILETYPE=BEANIPAD–National Income and Product Accounts Diskette Format

The diskette format National Income and Product Accounts files contain the same information as the tape format files described previously.

Data Files	Database is stored in a single diskette file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	
BY variables	PARTNO	Part Number of Publication,

		Integer Portion of the Table Number, 1-9 (character)
	TABNUM	Table Number Within Part, Decimal Portion of the Table Number, 1-24 (character)
Series Variables	Series variable names are constructed by concatenating table number suffix, line and column numbers within each table. An underscore (_) prefix is also added for readability.	

BLS Data Files

The Bureau of Labor Statistics, U.S. Department of Labor, compiles and distributes data on employment, expenditures, prices, productivity, injuries and illnesses, and wages.

The following BLS file types are supported:

FILETYPE=BLSCPI–Consumer Price Index Surveys (=CU,CW)

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR, SEMIYEAR1.6, MONTH (default)	
BY variables	SURVEY	Survey type: CU=All Urban Consumers, CW=Urban Wage Earners and Clerical Workers (character)
	SEASON	Seasonality: S=Seasonally adjusted, U=Unadjusted (character)
	AREA	Geographic Area (character)
	BASPTYPE	Index Base Period Type, S=Standard, A=Alternate Reference (character)
	BASEPER	Index Base Period (character)
Series Variables	Series variable names are the same as consumer item codes listed in the Series Directory shipped with the data tapes.	
Missing Codes	A data value of 0 is interpreted as MISSING.	

FILETYPE=BLSWPI–Producer Price Index Survey (WP)

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR, MONTH (default)	
BY variables	SEASON	Seasonality: S=Seasonally adjusted, U=Unadjusted (character)
	MAJORCOM	Major Commodity Group (character)
Sorting Order	BY SEASON MAJORCOM	

Series Variables	Series variable names are the same as commodity codes but prefixed by an underscore (_).
Missing Codes	A data value of 0 is interpreted as MISSING.

FILETYPE=BLSEENA–National Employment, Hours, and Earnings Survey

Data Files	Database is stored in a single tape file.	
INTERVAL=	YEAR, QUARTER, MONTH (default)	
BY variables	SEASON	Seasonality: S=Seasonally adjusted, U=Unadjusted (character)
	DIVISION	Major Industrial Division (character)
	INDUSTRY	Industry Code (character)
Sorting Order	BY SEASON DIVISION INDUSTRY	
Series Variables	Series variable names are the same as data type codes prefixed by EE.	
	EE01	Total Employment,
	EE02	Employment of Women
	EE03	Employment of Production or Nonsupervisory Workers
	EE04	Average Weekly Earnings of Production Workers
	EE05	Average Weekly Hours of Production Workers
	EE06	Average Hourly Earnings of Production Workers
	EE07	Average Weekly Overtime Hours of Production Workers
	EE40	Index of Aggregate Weekly Hours
	EE41	Index of Aggregate Weekly Payrolls
	EE47	Hourly Earnings Index; 1977 Weights; Current Dollars
	EE48	Hourly Earnings Index; 1977 Weights; Base 1977 Dollars
	EE49	Average Hourly Earnings; Base 1977 Dollars
	EE50	Gross Average Weekly Earnings; Current Dollars
	EE51	Gross Average Weekly Earnings; Base 1977 Dollars
	EE52	Spendable Average Weekly Earnings; No Dependents; Current Dollars
	EE53	Spendable Average Weekly Earnings; No Dependents; Base 1977 Dollars
	EE54	Spendable Average Weekly Earnings; 3 Dependents; Current Dollars
	EE55	Spendable Average Weekly Earnings; 3 Dependents; Base 1977 Dollars
	EE60	Average Hourly Earnings Excluding Overtime
	EE61	Index of Diffusion; 1-month Span; Base 1977

	EE62	Index of Diffusion; 3-month Span; Base 1977
	EE63	Index of Diffusion; 6-month Span; Base 1977
	EE64	Index of Diffusion; 12-month Span; Base 1977
Missing Codes	Series data values are set to MISSING when their status codes are 1.	

FILETYPE=BLSEESA–State and Area Employment, Hours, and Earnings Survey

Data Files	Database is stored in a single tape file.	
INTERVAL=	YEAR, MONTH (default)	
BY variables	STATE	State FIPS codes (numeric)
	AREA	Area Codes (character)
	DIVISION	Major Industrial Division (character)
	INDUSTRY	Industry Code (character)
	DETAIL	Private/Government Detail
Sorting Order	BY STATE AREA DIVISION INDUSTRY DETAIL	
Series Variables	Series variable names are the same as data type codes prefixed by SA.	
	SA1	All employees
	SA2	Women workers
	SA3	Production Workers
	SA4	Average weekly earnings
	SA5	Average weekly hours
Missing Codes	Series data values are set to MISSING when their status codes are 1.	

DRI/McGraw-Hill Data Files

The DRIBASIC (formerly CITIBASE) database contains economic and financial indicators of the U.S. and international economies gathered from various government and private sources by DRI/McGraw-Hill, Inc. There are over 8000 yearly, quarterly, monthly, weekly, and daily time series.

DRI/McGraw-Hill distributes Basic Economic data files on various media. DRI also offers Data Delivery Service (DDS)data files via DRIPRO's data retrieval software called Xtract. Most DDS data files can be read by DATASOURCE using the DRIDDS filetype.

The following DRI file types are supported:

FILETYPE=DRIBASIC–DRI Basic Economic Data Files

Data Files	Database is stored in a single file.
INTERVAL=	YEAR (default), QUARTER, MONTH, WEEK, WEEK1.1, WEEK1.2, WEEK1.3, WEEK1.4, WEEK1.5, WEEK1.6, WEEK1.7, WEEKDAY
BY variables	None
Series Variables	Variable names are taken from the series descriptor records in the data file . Note that series codes can be 20 bytes.
Missing Codes	MISSING=('1.000000E9'=. 'NA'-'ND'=.)

Note that when you specify the INTERVAL=WEEK option, all the weekly series will be aggregated, and the DATE variable in the OUT= data set will be set to the date of Sundays. The date of first observation for each series is the Sunday marking the beginning of the week that contains the starting date of that variable.

FILETYPE=DRIDDS–DRI Data Delivery Service Data Files

Data Files	Database is stored in a single file.
INTERVAL=	YEAR (default), SEMIYEAR, QUARTER, MONTH, SEMI-MONTH, TENDAY, WEEK, WEEK1.1, WEEK1.2, WEEK1.3, WEEK1.4, WEEK1.5, WEEK1.6, WEEK1.7, WEEKDAY, DAY
BY variables	None
Series Variables	Variable names are taken from the series descriptor records in the data file . Note that series names can be 24 bytes.
Missing Codes	MISSING=('NA'-'ND'=.)

FILETYPE=CITIOLD–Old format CITIBASE data files

This file type is used for CITIBASE data tapes distributed prior to May, 1987.

Data Files	Database is stored in a single file.
INTERVAL=	YEAR (default), QUARTER, MONTH
BY variables	None
Series Variables	Variable names are taken from the series descriptor records in the data file and are the same as the series codes reported in the *CITIBASE Directory*.
Missing Codes	1.0E9=.

FILETYPE=CITIDISK–PC Diskette Format CITIBASE Databases

Data Files	Database is stored in groups of three associated files having the same file name but different extensions: KEY, IND, or DB. The INFILE= option should contain three filerefs in the following order: INFILE=(*keyfile indfile dbfile*)
INTERVAL=	YEAR (default), QUARTER, MONTH
BY variables	None
Series Variables	Series variable names are the same as series codes reported in the *CITIBASE Directory*.
Missing Codes	1.0E9=.

COMPUSTAT Data Files

COMPUSTAT data files, distributed by Standard and Poor's Compustat Services, Inc., consist of a collection of financial, statistical, and market information covering several thousand industrial and nonindustrial companies. Data are available in both an IBM 360/370 format and a "Universal Character" format, both of which further subdivide into annual and quarterly formats.

The BY variables are used to select individual companies or a group of companies. Individual companies can be selected by their unique six-digit CUSIP issuer code (CNUM). A number of specific groups of companies can be extracted from the tape by the following key fields:

FILE	specifies the file identification code used to group companies by files
ZLIST	specifies the exchange listing code that can be used to group companies by exchange
DNUM	is used to extract companies in a specific SIC industry group

Series names are internally constructed from the data array names documented in the COMPUSTAT manual. Each column of data array is treated as a SAS variable. The names of these variables are generated by concatenating the corresponding column numbers to the array name.

Missing values use four codes. Missing code '.C' represents a combined figure where the data item has been combined into another data item, '.I' reports an insignificant figure, '.S' represents a semi-annual figure in the second and fourth quarters, '.A' represents an annual figure in the fourth quarter, and '.' indicates that the data item is not available. The missing codes '.C' and '.I' are not used for Aggregate or Prices, Dividends, and Earnings (PDE) files. The missing codes '.S' and '.A' are used only on the Industrial Quarterly File and not on the Aggregate Quarterly, Business Information, or PDE files.

FILETYPE=CSAIBM–COMPUSTAT Annual, IBM 360/370 Format

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default)	
BY variables	DNUM	Industry Classification Code (numeric)
	CNUM	CUSIP Issuer Code (character)
	CIC	CUSIP Issue Number and Check Digit (numeric)
	FILE	File Identification Code (numeric)
	ZLIST	Exchange Listing and S&P Index Code (numeric)
	CONAME	Company Name (character)
	INAME	Industry Name (character)
	SMBL	Stock Ticker Symbol (character)
	XREL	S&P Industry Index Relative Code (numeric)
	STK	Stock Ownership Code (numeric)
	STATE	Company Location Identification Code - State (numeric)
	COUNTY	Company Location Identification Code - County (numeric)
	FINC	Incorporation Code - Foreign (numeric)
	EIN	Employer Identification Number (character)
	CPSPIN	S&P Index Primary Marker (character)
	CSSPIN	S&P Index Secondary Identifier (character)
	CSSPII	S&P Index Subset Identifier (character)
	SDBT	S&P Senior Debt Rating - Current (character)
	SDBTIM	Footnote- S&P Senior Debt Rating- Current (character)
	SUBDBT	S&P Subordinated Debt Rating - Current (character)
	CPAPER	S&P Commercial Paper Rating - Current (character)
Sorting order	BY DNUM CNUM CIC	
Series Variables	DATA1-DATA350 FYR UCODE SOURCE AFTNT1-AFTNT70	
Default KEEP List	DROP DATA321-DATA326 DATA337-DATA350 AFTNT51-AFTNT70;	
Missing Codes	0.0001=. 0.0004=.C 0.0008=.I 0.0002=.S 0.0003=.A	

FILETYPE=CS48QIBM–COMPUSTAT 48-Quarter, IBM 360/370 Format

Data Files	Database is stored in a single file.	
INTERVAL=	QUARTER (default)	
BY variables	DNUM	Industry Classification Code (numeric)
	CNUM	CUSIP Issuer Code (character)
	CIC	CUSIP Issue Number and Check Digit (numeric)
	FILE	File Identification Code (numeric)

	CONAME	Company Name (character)
	INAME	Industry Name (character)
	EIN	Employer Identification Number (character)
	STK	Stock Ownership Code (numeric)
	SMBL	Stock Ticker Symbol (character)
	ZLIST	Exchange Listing and S&P Index Code (numeric)
	XREL	S&P Industry Index Relative Code (numeric)
	FIC	Incorporation Code - Foreign (numeric)
	INCORP	Incorporation Code - State (numeric)
	STATE	Company Location Identification Code - State (numeric)
	COUNTY	Company Location Identification Code - County (numeric)
	CANDX	Canadian Index Code - Current (character)
Sorting order	BY DNUM CNUM CIC;	
Series Variables	DATA1-DATA232	Data Array
	QFTNT1-QFTNT60	Data Footnotes
	FYR	Fiscal Year-end Month of Data
	SPCSCYR	SPCS Calendar Year
	SPCSCQTR	SPCS Calendar Quarter
	UCODE	Update Code
	SOURCE	Source Document Code
	BONDRATE	S&P Bond Rating
	DEBTCL	S&P Class of Debt
	CPRATE	S&P Commercial Paper Rating
	STOCK	S&P Common Stock Ranking
	MIC	S&P Major Index Code
	IIC	S&P Industry Index Code
	REPORTDT	Report Date of Quarterly Earnings
	FORMAT	Flow of Funds Statement Format Code
	DEBTRT	S&P Subordinated Debt Rating
	CANIC	Canadian Index Code
	CS	Comparability Status
	CSA	Company Status Alert
	SENIOR	S&P Senior Debt Rating
Default KEEP List	DROP DATA122-DATA232 QFTNT24-QFTNT60;	
Missing Codes	0.0001=. 0.0004=.C 0.0008=.I 0.0002=.S 0.0003=.A	

FILETYPE=CSAUC–COMPUSTAT Annual, Universal Character Format

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default)	
BY variables	DNUM	Industry Classification Code (numeric)
	CNUM	CUSIP Issuer Code (character)
	CIC	CUSIP Issue Number and Check Digit (character)
	FILE	File Identification Code (numeric)
	ZLIST	Exchange Listing and S&P Index Code (numeric)
	CONAME	Company Name (character)
	INAME	Industry Name (character)
	SMBL	Stock Ticker Symbol (character)
	XREL	S&P Industry Index Relative Code (numeric)
	STK	Stock Ownership Code (numeric)
	STATE	Company Location Identification Code - State (numeric)
	COUNTY	Company Location Identification Code - County (numeric)
	FINC	Incorporation Code - Foreign (numeric)
	EIN	Employer Identification Number (character)
	CPSPIN	S&P Index Primary Marker (character)
	CSSPIN	S&P Index Secondary Identifier (character)
	CSSPII	S&P Index Subset Identifier (character)
	SDBT	S&P Senior Debt Rating - Current (character)
	SDBTIM	Footnote- S&P Senior Debt Rating- Current (character)
	SUBDBT	S&P Subordinated Debt Rating - Current (character)
	CPAPER	S&P Commercial Paper Rating - Current (character)
Sorting order	BY DNUM CNUM CIC	
Series Variables	DATA1-DATA350 FYR UCODE SOURCE AFTNT1-AFTNT70	
Default KEEP List	DROP DATA321-DATA326 DATA337-DATA350 AFTNT51-AFTNT70;	
Missing Codes	-0.001=. -0.004=.C -0.008=.I -0.002=.S -0.003=.A	

FILETYPE=CS48QUC–COMPUSTAT 48 Quarter, Universal Character Format

Data Files	Database is stored in a single file.	
INTERVAL=	QUARTER (default)	
BY variables	DNUM	Industry Classification Code (numeric)
	CNUM	CUSIP Issuer Code (character)
	CIC	CUSIP Issue Number and Check Digit (character)
	FILE	File Identification Code (numeric)

	CONAME	Company Name (character)
	INAME	Industry Name (character)
	EIN	Employer Identification Number (character)
	STK	Stock Ownership Code (numeric)
	SMBL	Stock Ticker Symbol (character)
	ZLIST	Exchange Listing and S&P Index Code (numeric)
	XREL	S&P Industry Index Relative Code (numeric)
	FIC	Incorporation Code - Foreign (numeric)
	INCORP	Incorporation Code - State (numeric)
	STATE	Company Location Identification Code - State (numeric)
	COUNTY	Company Location Identification Code - County (numeric)
	CANDXC	Canadian Index Code - Current (numeric)
Sorting order	BY DNUM CNUM CIC	
Series Variables	DATA1-DATA232	Data Array
	QFTNT1-QFTNT60	Data Footnotes
	FYR	Fiscal Year-end Month of Data
	SPCSCYR	SPCS Calendar Year
	SPCSCQTR	SPCS Calendar Quarter
	UCODE	Update Code
	SOURCE	Source Document Code
	BONDRATE	S&P Bond Rating
	DEBTCL	S&P Class of Debt
	CPRATE	S&P Commercial Paper Rating
	STOCK	S&P Common Stock Ranking
	MIC	S&P Major Index Code
	IIC	S&P Industry Index Code
	REPORTDT	Report Date of Quarterly Earnings
	FORMAT	Flow of Funds Statement Format Code
	DEBTRT	S&P Subordinated Debt Rating
	CANIC	Canadian Index Code - Current
	CS	Comparability Status
	CSA	Company Status Alert
	SENIOR	S&P Senior Debt Rating
Default KEEP List	DROP DATA122-DATA232 QFTNT24-QFTNT60;	
Missing Codes	-0.001=. -0.004=.C -0.008=.I -0.002=.S -0.003=.A	

CRSP Stock Files

The Center for Research in Security Prices provides comprehensive security price data via two primary stock files, the NYSE/AMEX file and the NASDAQ file. These

files are composed of master and return components, available separately or combined. CRSP stock files are further differentiated by the frequency at which prices and returns are reported, daily or monthly. Both daily and monthly files contain annual data fields.

CRSP data files come either in binary or character tape format, or in CRSP Access97 CDROM format.

CRSP stock data are provided in two files, a main data file containing security information and a calendar/indices file containing a list of trading dates and market information associated with those trading dates. If security data do not fit on one tape, they are split into two or more files, each one of which resides on a different self-contained tape. The calendar/indices file is on the first tape only.

The file types for CRSP stock files are constructed by concatenating CRSP with a D or M to indicate the frequency of data, followed by B,C, or I to indicate file formats. B is for host binary, C is for character, and I is for IBM binary formats. The last character in the file type indicates if you are reading the Calendar/Indices file (I), or if you are extracting the security (S) or annual data (A). For example, the file type for the daily NYSE/AMEX combined tape in IBM binary format is CRSPDIS. Its calendar/indices file can be read by CRSPDII, and its annual data can be extracted by CRSPDIA.

Starting in 1995, binary data tapes use split records (RICFAC=2) so the 1995 filetypes (CR95*) should be used for 1995 and 1996 binary data.

If you use utility routines supplied by CRSP to convert a character format file to a binary format file on a given host, then you need to use host binary file types (RIDFAC=1) to read those files in. Note that you can not do the conversion on one host and transfer and read the file on another host.

If you are using the CRSP Access97 Database, you will need to use the utility routine (stk_dump_bin) supplied by CRSP to generate the UNIX binary format of the data. You can access the UNIX (or SUN) binary data by using PROC DATASOURCE with the CRSPDUS for daily or CRSPMUS for monthly stock data. See the example on Example 10.11 later in this chapter.

For CRSP file types, the INFILE= option must be of the form

```
INFILE=( calfile security1 < security2 ... > )
```

where *calfile* is the fileref assigned to the calendar/indices file, and *securty1 < securty2 ... >* are the filerefs given to the security files, in the order in which they should be read.

CRSP Calendar/Indices Files

Data Files	Database is stored in a single file.	
INTERVAL=	DAY	for products DA, DR, DX, EX, NX and RA
	MONTH	for products MA, MX and MZ

BY variables	None	
Series Variables	VWRETD	Value-Weighted Return (including all distributions)
	VWRETX	Value-Weighted Return (excluding dividends)
	EWRETD	Equal-Weighted Return (including all distributions)
	EWRETX	Equal-Weighted Return (excluding dividends)
	TOTVAL	Total Market Value
	TOTCNT	Total Market Count
	USDVAL	Market Value of Securities Used
	USDCNT	Count of Securities Used
	SPINDX	Level of the Standard & Poor's Composite Index
	SPRTRN	Return on the Standard & Poor's Composite Index
	NCINDX	NASDAQ Composite Index
	NCRTRN	NASDAQ Composite Return
Default KEEP List	All variables will be kept.	

CRSP Daily Security Files

Data Files	INFILE=(calfile securty1 < securty2 ... >)		
INTERVAL=	DAY		
BY variables	CUSIP	CUSIP Identifier (character)	
	PERMNO	CRSP Permanent Number (numeric)	
	COMPNO	NASDAQ Company Number (numeric)	
	ISSUNO	NASDAQ Issue Number (numeric)	
	HEXCD	Header Exchange Code (numeric)	
	HSICCD	Header SIC Code (numeric)	
Sorting Order	BY CUSIP		
Series Variables	BIDLO	Bid or Low	
	ASKHI	Ask or High	
	PRC	Closing Price of Bid/Ask Average	
	VOL	Share Volume	
	RET	Holding Period Return missing=(-66.0 = .p -77.0 = .t -88.0 = .r -99.0 = .b)	
	BXRET	Beta Excess Return missing=(-44.0 = .)	
	SXRET	Standard Deviation Excess Return missing=(-44.0 = .)	
Events	NAMES	NCUSIP	Name CUSIP
		TICKER	Exchange Ticker Symbol
		COMNAM	Company Name
		SHRCLS	Share Class

		SHRCD	Share Code
		EXCHCD	Exchange Code
		SICCD	Standard Industrial Classification Code
	DIST	DISTCD	Distribution Code
		DIVAMT	Dividend Cash Amount
		FACPR	Factor to Adjust Price
		FACSHR	Factor to Adjust Shares Outstanding
		DCLRDT	Declaration Date
		RCRDDT	Record Date
		PAYDT	Payment Date
	SHARES	SHROUT	Number of Shares Outstanding
		SHRFLG	Share Flag
	DELIST	DLSTCD	Delisting Code
		NWPERM	New CRSP Permanent Number
		NEXTDT	Date of Next Available Information
		DLBID	Delisting Bid
		DLASK	Delisting Ask
		DLPRC	Delisting Price
		DLVOL	Delisting Volume missing=(-99 = .)
		DLRET	Delisting Return missing=(-55.0=.s -66.0=.t -88.0=.a -99.0=.p);
	NASDIN	TRTSCD	Traits Code
		NMSIND	National Market System Indicator
		MMCNT	Market Maker Count
		NSDINX	NASD Index
Default KEEP Lists	All periodic series variables will be output to the OUT= data set and all event variables will be output to the OUTEVENT= data set.		

CRSP Monthly Security Files

Data Files	INFILE=(calfile securty1 < securty2 ... >)	
INTERVAL=	MONTH	
BY variables	CUSIP	CUSIP Identifier (character)
	PERMNO	CRSP Permanent Number (numeric)
	COMPNO	NASDAQ Company Number (numeric)
	ISSUNO	NASDAQ Issue Number (numeric)
	HEXCD	Header Exchange Code (numeric)
	HSICCD	Header SIC Code (numeric)

Sorting Order	BY CUSIP		
Series Variables	BIDLO	Bid or Low	
	ASKHI	Ask or High	
	PRC	Closing Price of Bid/Ask average	
	VOL	Share Volume	
	RET	Holding Period Return missing=(-66.0 = .p -77.0 = .t -88.0 = .r -99.0 = .b);	
	RETX	Return Without Dividends missing=(-44.0 = .)	
	PRC2	Secondary Price missing=(-44.0 = .)	
Events	NAMES	NCUSIP	Name CUSIP
		TICKER	Exchange Ticker Symbol
		COMNAM	Company Name
		SHRCLS	Share Class
		SHRCD	Share Code
		EXCHCD	Exchange Code
		SICCD	Standard Industrial Classification Code
	DIST	DISTCD	Distribution Code
		DIVAMT	Dividend Cash Amount
		FACPR	Factor to Adjust Price
		FACSHR	Factor to Adjust Shares Outstanding
		EXDT	Ex-distribution Date
		RCRDDT	Record Date
		PAYDT	Payment Date
	SHARES	SHROUT	Number of Shares Outstanding
		SHRFLG	Share Flag
	DELIST	DLSTCD	Delisting Code
		NWPERM	New CRSP Permanent Number
		NEXTDT	Date of Next Available Information
		DLBID	Delisting Bid
		DLASK	Delisting Ask
		DLPRC	Delisting Price
		DLVOL	Delisting Volume
		DLRET	Delisting Return missing=(-55.0=.s -66.0=.t -88.0=.a -99.0=.p);
	NASDIN	TRTSCD	Traits Code
		NMSIND	National Market System Indicator
		MMCNT	Market Maker Count
		NSDINX	NASD Index

Default Lists	KEEP	All periodic series variables will be output to the OUT= data set and all event variables will be output to the OUTEVENT= data set.

CRSP Annual Data

Data Files	INFILE=(securty1 < securty2 ... >)	
INTERVAL=	YEAR	
BY variables	CUSIP	CUSIP Identifier (character)
	PERMNO	CRSP Permanent Number (numeric)
	COMPNO	NASDAQ Company Number (numeric)
	ISSUNO	NASDAQ Issue Number (numeric)
	HEXCD	Header Exchange Code (numeric)
	HSICCD	Header SIC Code (numeric)
Sorting Order	BY CUSIP	
Series Variables	CAPV	Year End Capitalization
	SDEVV	Annual Standard Deviation missing=(-99.0 = .)
	BETAV	Annual Beta missing=(-99.0 = .)
	CAPN	Year End Capitalization Portfolio Assignment
	SDEVN	Standard Deviation Portfolio Assignment
	BETAN	Beta Portfolio Assignment
Default Lists	KEEP	All variables will be kept.

FAME Information Services Databases

The DATASOURCE procedure provides access to FAME Information Services databases for Unix-based systems only. For a more flexible FAME Data Base access use the SASEFAME interface engine, see Chapter 5, “The SASEFAME Interface Engine,”, which is supported on Windows NT, Solaris2, AIX, and HP-UX hosts.

The DATASOURCE interface to FAME requires a component supplied by FAME Information Services, Inc. Once this FAME component is installed on your system, you can use the DATASOURCE procedure to extract data from your FAME databases as follows:

- Specify FILETYPE=FAME on the PROC DATASOURCE statement.
- Specify the FAME database to access with a DBNAME=’*fame-database*’ option on the PROC DATASOURCE statement. The character string you specify on the DBNAME= option is passed through to FAME; specify the value of this option as you would in accessing the database from within FAME software.

- Specify the output SAS data set to be created, the frequency of the series to be extracted, and other usual DATASOURCE procedure options as appropriate.
- Specify the time range to extract with a RANGE statement. The RANGE statement is required when extracting series from FAME databases.
- Specify the FAME series to be extracted with a KEEP statement. The items on the KEEP statement are passed through to FAME software; therefore, you can use any valid FAME expression to specify the series to be extracted. Put in quotes any FAME series name or expression that is not a valid SAS name.
- Specify the SAS variable names you want to use for the extracted series on a RENAME statement. Give the FAME series name or expression (in quotes if needed) followed by an equal sign and the SAS name. The RENAME statement is not required; however, if the FAME series name is not a valid SAS variable name, the DATASOURCE procedure will construct a SAS name by translating and truncating the FAME series name. This process may not produce the desired name for the variable in the output SAS data set, so a rename statement could be used to produce a more appropriate variable name. The VALIDVARNAME=ANY option on your SAS options statement can be used to allow special characters in the SAS variable name.

For an alternative solution to PROC DATASOURCE's access to FAME, see the chapter on the SASEFAME Interface Engine.

FILETYPE=FAME–FAME Information Services Databases

INTERVAL=	YEAR	corresponds to FAME's ANNUAL(DECEMBER)
	YEAR.2	correspond to FAME's ANNUAL(JANUARY)
	YEAR.3	correspond to FAME's ANNUAL(FEBRUARY)
	YEAR.4	correspond to FAME's ANNUAL(MARCH)
	YEAR.5	correspond to FAME's ANNUAL(APRIL)
	YEAR.6	correspond to FAME's ANNUAL(MAY)
	YEAR.7	correspond to FAME's ANNUAL(JUNE)
	YEAR.8	correspond to FAME's ANNUAL(JULY)
	YEAR.9	correspond to FAME's ANNUAL(AUGUST)
	YEAR.10	correspond to FAME's ANNUAL(SEPTEMBER)
	YEAR.11	correspond to FAME's ANNUAL(OCTOBER)
	YEAR.12	correspond to FAME's ANNUAL(NOVEMBER)
	SEMIYEAR, QUARTER, MONTH, SEMIMONTH, TENDAY	are supported frequencies
	WEEK	corresponds to FAME's WEEKLY(SATURDAY)
	WEEK.2	corresponds to FAME's WEEKLY(SUNDAY)
	WEEK.3	corresponds to FAME's WEEKLY(MONDAY)
	WEEK.4	corresponds to FAME's WEEKLY(TUESDAY)

	WEEK.5	corresponds to FAME's WEEKLY(WEDNESDAY)
	WEEK.6	corresponds to FAME's WEEKLY(THURSDAY)
	WEEK.7	corresponds to FAME's WEEKLY(FRIDAY)
	WEEK2	corresponds to FAME's BIWEEKLY(ASATURDAY)
	WEEK2.2	correspond to FAME's BIWEEKLY(ASUNDAY)
	WEEK2.3	correspond to FAME's BIWEEKLY(AMONDAY)
	WEEK2.4	correspond to FAME's BIWEEKLY(ATUESDAY)
	WEEK2.5	correspond to FAME's BIWEEKLY(AWEDNESDAY)
	WEEK2.6	correspond to FAME's BIWEEKLY(ATHURSDAY)
	WEEK2.7	correspond to FAME's BIWEEKLY(AFRIDAY)
	WEEK2.8	correspond to FAME's BIWEEKLY(BSATURDAY)
	WEEK2.9	correspond to FAME's BIWEEKLY(BSUNDAY)
	WEEK2.10	correspond to FAME's BIWEEKLY(BMONDAY)
	WEEK2.11	correspond to FAME's BIWEEKLY(BTUESDAY)
	WEEK2.12	correspond to FAME's BIWEEKLY(BWEDNESDAY)
	WEEK2.13	correspond to FAME's BIWEEKLY(BTHURSDAY)
	WEEK2.14	correspond to FAME's BIWEEKLY(BFRIDAY)
	WEEKDAY, DAY	are supported frequencies
BY variables	None	
Series Variables	Variable names are constructed from the FAME series codes. Note that series names are limited to 32 bytes.	

Haver Analytics Data Files

Haver Analytics offers a broad range of economic, financial, and industrial data for the U.S. and other countries. The format of Haver Analytics data files is similar to the CITIBASE format.

FILETYPE=HAVER–Haver Analytics Data Files HAVERO–Old format Haver Files

Data Files	Database is stored in a single file.
INTERVAL=	YEAR (default), QUARTER, MONTH

BY variables	1.0E9=.
Series Variables	Variable names are taken from the series descriptor records in the data file. NOTE: HAVER filetype reports the UPDATE and SOURCE in the OUTCONT= data set, while HAVERO does not.
Missing Codes	1.0E9=.

IMF Data Files

The International Monetary Fund's Economic Information System (EIS) offers tape subscriptions for their International Financial Statistics (IFS), Direction of Trade Statistics (DOTS), Balance of Payment Statistics (BOPS), and the Government Finance Statistics (GFS) databases. The first three contain annual, quarterly, and monthly data, while the GFS file has only annual data.

IMF data tapes are available for IBM mainframe systems (EBCDIC character coding) in both a "packed" and an "unpacked" format. PROC DATASOURCE supports only the "packed" format at this time.

FILETYPE=IMFIFSP–International Financial Statistics, Packed format

The IFS data files contain over 23,000 time series including interest and exchange rates, national income and product accounts, price and production indexes, money and banking, export commodity prices, and balance of payments for nearly 200 countries and regional aggregates.

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	
BY variables	COUNTRY	Country Code (character, three-digits)
	CSC	Control Source Code (character)
	PARTNER	Partner Country Code (character, three-digits)
	VERSION	Version Code (character)
Sorting Order	BY COUNTRY CSC PARTNER VERSION	
Series Variables	Series variable names are the same as series codes reported in *IMF Documentation* prefixed by F for data and F_F for footnote indicators.	
Default KEEP List	By default all the footnote indicators will be dropped.	

FILETYPE=IMFDOTSP–Direction of Trade Statistics, Packed Format

The DOTS files contain time series on the distribution of exports and imports for about 160 countries and country groups by partner country and areas.

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	
BY variables	COUNTRY	Country Code (character, three-digits)
	CSC	Control Source Code (character)
	PARTNER	Partner Country Code (character, three-digits)
	VERSION	Version Code (character)
Sorting Order	BY COUNTRY CSC PARTNER VERSION	
Series Variables	Series variable names are the same as series codes reported in *IMF Documentation* prefixed by D for data and F_D for footnote indicators.	
Default KEEP List	By default all the footnote indicators will be dropped.	

FILETYPE=IMFBOPSP–Balance of Payment Statistics, Packed Format

The BOPS data files contain approximately 43,000 time series on balance of payments for about 120 countries.

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	
BY variables	COUNTRY	Country Code (character, three-digits)
	CSC	Control Source Code (character)
	PARTNER	Partner Country Code (character, three-digits)
	VERSION	Version Code (character)
Sorting Order	BY COUNTRY CSC PARTNER VERSION	
Series Variables	Series variable names are the same as series codes reported in *IMF Documentation* prefixed by B for data and F_B for footnote indicators.	
Default KEEP List	By default all the footnote indicators will be dropped.	

FILETYPE=IMFGFSP–Government Finance Statistics, Packed Format

The GFS data files encompass approximately 28,000 time series that give a detailed picture of federal government revenue, grants, expenditures, lending minus repayment financing and debt, and summary data of state and local governments, covering 128 countries.

Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	
BY variables	COUNTRY	Country Code (character, three-digits)
	CSC	Control Source Code (character)
	PARTNER	Partner Country Code (character, three-digits)
	VERSION	Version Code (character)
Sorting Order	BY COUNTRY CSC PARTNER VERSION	
Series Variables	Series variable names are the same as series codes reported in *IMF Documentation* prefixed by G for data and F_G for footnote indicators.	
Default KEEP List	By default all the footnote indicators will be dropped.	

OECD Data Files

The Organization for Economic Cooperation and Development compiles and distributes statistical data, including National Accounts and Main Economic Indicators.

FILETYPE=OECDANA–Annual National Accounts

The ANA data files contain both main national aggregates accounts (Volume I) and detailed tables for each OECD Member country (Volume II).

Data Files	Database is stored on a single tape file.	
INTERVAL=	YEAR (default), SEMIYR1.6, QUARTER, MONTH, WEEK, WEEKDAY	
BY variables	PREFIX	Table number prefix (character)
	CNTRYZ	Country Code (character)
Series Variables	Series variable names are the same as the mnemonic name of the element given on the element 'E' record. They are taken from the 12 byte time series 'T' record time series indicative.	

```
rename p0discgdpe=p0digdpe;
rename doll2gdpe=dol2gdpe;
rename doll3gdpe=dol3gdpe;
rename doll1gdpe=dol1gdpe;
rename ppp1gdpd=pp1gdpd;
rename ppp1gdpd1=pp1gdpd1;
rename p0itxgdpc=p0itgdpc;
rename p0itxgdps=p0itgdps;
rename p0subgdpc=p0sugdpc;
rename p0subgdps=p0sugdps;
rename p0cfcgdpc=p0cfgdpc;
rename p0cfcgdps=p0cfgdps;
rename p0discgdpc=p0dicgdc;
rename p0discgdps=p0dicgds;
```

Missing Codes	A data value of * is interpreted as MISSING.

FILETYPE=OECDQNA–Quarterly National Accounts

The QNA file contains the main aggregates of quarterly national accounts for 16 OECD Member Countries and on a selected number of aggregates for 4 groups of member countries: OECD-Total, OECD-Europe, EEC, and the 7 major countries.

Data Files	Database is stored on a single file.	
INTERVAL=	QUARTER(default),YEAR	
BY variables	COUNTRY	Country Code (character)
	SEASON	Seasonality S=Seasonally adjusted 0=raw data, not seasonally adjusted
	PRICETAG	Prices C=data at current prices R,L,M=data at constant prices P,K,J,V=implicit price index or volume index
Series Variables	Subject code used to distinguish series within countries. Series variables are prefixed by _ for data, C for control codes, and D for relative date.	
Default DROP List	By default all the control codes and relative dates will be dropped.	
Missing Codes	A data value of + or - is interpreted as MISSING.	

FILETYPE=OECDMEI–Main Economic Indicators

The MEI file contains all series found in Parts 1 and 2 of the publication *Main Economic Indicators.*

Data Files	Database is stored on a single file.	
INTERVAL=	YEAR(default),QUARTER,MONTH	
BY variables	COUNTRY	Country Code (character)
	CURRENCY	Unit of expression of the series.
	ADJUST	Adjustment 0,H,S,A,L=no adjustment 1,I=calendar or working day adjusted 2,B,J,M=seasonally adjusted by National Authorities 3,K,D=seasonally adjusted by OECD
Series Variables	Series variables are prefixed by _ for data, C for control codes, and D for relative date in weeks since last updated.	
Default DROP List	By default, all the control codes and relative dates will be dropped.	
Missing Codes	A data value of + or - is interpreted as MISSING.	

Examples

Example 10.1. BEA National Income and Product Accounts

In this example, exports and imports of goods and services are extracted to demonstrate how to work with a National Income and Product Accounts Tape file.

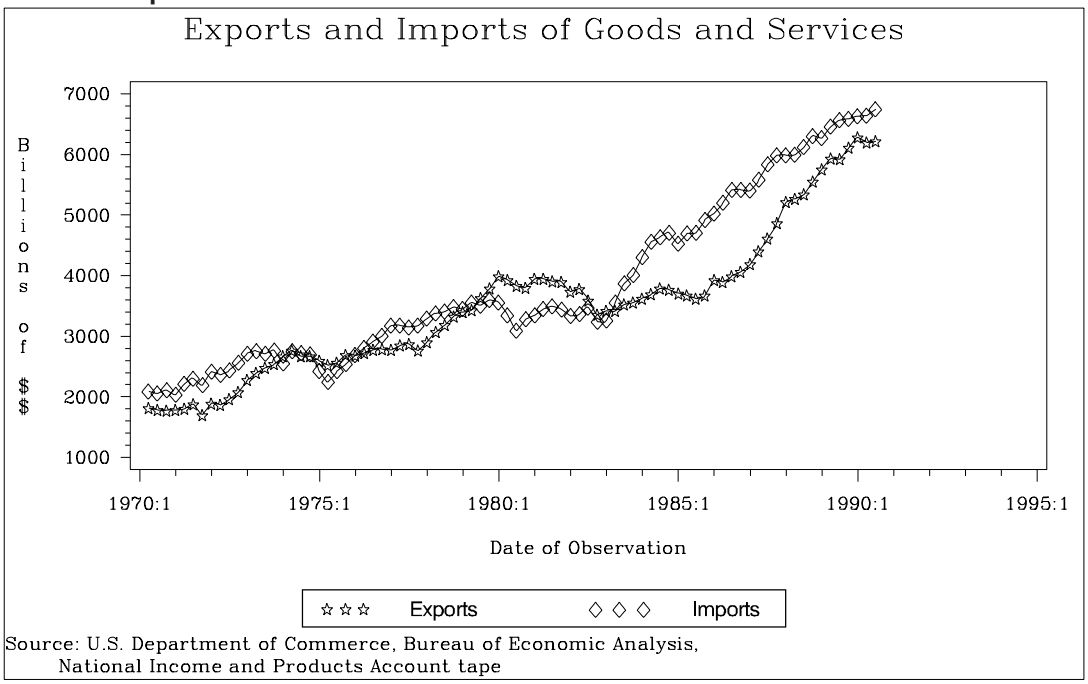
From the "Statistical Tables" published by the United States Department of Commerce, Bureau of Economic Analysis, exports and imports of goods and services are given in the second table (TABNUM='02') of the "Foreign Transactions" section (PARTNO='4'). This table does not have any table suffix A or B. Moreover, the first line in the table gives exports, while the eighth gives imports. Therefore, the series names for exports and imports are __00100 and __00800, where the first underscore is inserted by the procedure, the second underscore is the place holder for the table suffix, the following three digits are the line numbers, and the last two digits are the column numbers.

The following statements put this information together to extract quarterly exports and imports from a BEANIPA type file:

```
filename  datafile 'host-specific-path-name' host-options;
proc datasource filetype=beanipa infile=datafile
                interval=qtr out=foreign;
   keep __00100 __00800;
   where partno='4' and tabnum='02';
   label __00100='Exports of Goods and Services';
   label __00800='Imports of Goods and Services';
   rename __00100=exports __00800=imports;
run;
```

The plot of EXPORTS and IMPORTS against DATE is shown in Output 10.1.1.

Output 10.1.1. Plot of Time Series in the OUT= Data Set for FILETYPE=BEANIPA



This example illustrates the following features:

- You need to know the series variables names used by a particular vendor in order to construct the KEEP statement.
- You need to know the BY variable names and their values for the required cross sections.
- You can use RENAME and LABEL statements to associate more meaningful names and labels with your selected series variables.

Example 10.2. BLS Consumer Price Index Surveys

This example compares changes of the prices in medical care services with respect to different regions for all urban consumers (SURVEY='CU') since May, 1975. The source of data is the Consumer Price Index Surveys distributed by the U.S. Department of Labor, Bureau of Labor Statistics.

An initial run of PROC DATASOURCE gives the descriptive information on different regions available (the OUTBY= data set), as well as the series variable name corresponding to medical care services (the OUTCONT= data set).

```
filename datafile 'host-specific-file-name' <host-options>;
proc datasource filetype=blscpi interval=month
                outby=cpikey outcont=cpicont;
   where survey='CU';
run;

title1 'Partial Listing of the OUTBY= Data Set';
```

```
proc print data=cpikey noobs;
   where upcase(areaname) in
         ('NORTHEAST','NORTH CENTRAL','SOUTH','WEST');
run;

title1 'Partial Listing of the OUTCONT= Data Set';
proc print data=cpicont noobs;
   where index( upcase(label), 'MEDICAL CARE' );
run;
```

The OUTBY= data set in Output 10.2.1 lists all cross sections available for the four geographical regions: Northeast (AREA='0100'), North Central (AREA='0200'), Southern (AREA='0300'), and Western (AREA='0400'). The OUTCONT= data set gives the variable names for medical care related series.

Output 10.2.1. Partial Listings of the OUTBY= and OUTCONT= Data Sets

Partial Listing of the OUTBY= Data Set

survey	season	area	basptype	baseper	st_date	end_date
CU	U	0100	A	DECEMBER 1977=100	DEC1966	JUL1990
CU	U	0100	S	1982-84=100	DEC1966	JUL1990
CU	U	0100	S	DECEMBER 1982=100	DEC1982	JUL1990
CU	U	0100	S	DECEMBER 1986=100	DEC1986	JUL1990
CU	U	0200	A	DECEMBER 1977=100	DEC1966	JUL1990
CU	U	0200	S	1982-84=100	DEC1966	JUL1990
CU	U	0200	S	DECEMBER 1982=100	DEC1982	JUL1990
CU	U	0200	S	DECEMBER 1986=100	DEC1986	JUL1990
CU	U	0300	A	DECEMBER 1977=100	DEC1966	JUL1990
CU	U	0300	S	1982-84=100	DEC1966	JUL1990
CU	U	0300	S	DECEMBER 1982=100	DEC1982	JUL1990
CU	U	0300	S	DECEMBER 1986=100	DEC1986	JUL1990
CU	U	0400	A	DECEMBER 1977=100	DEC1966	JUL1990
CU	U	0400	S	1982-84=100	DEC1966	JUL1990
CU	U	0400	S	DECEMBER 1982=100	DEC1982	JUL1990
CU	U	0400	S	DECEMBER 1986=100	DEC1986	JUL1990

ntime	nobs	nseries	nselect	surtitle	areaname
284	284	1	1	ALL URBAN CONSUM	NORTHEAST
284	284	90	90	ALL URBAN CONSUM	NORTHEAST
92	92	7	7	ALL URBAN CONSUM	NORTHEAST
44	44	1	1	ALL URBAN CONSUM	NORTHEAST
284	284	1	1	ALL URBAN CONSUM	NORTH CENTRAL
284	284	90	90	ALL URBAN CONSUM	NORTH CENTRAL
92	92	7	7	ALL URBAN CONSUM	NORTH CENTRAL
44	44	1	1	ALL URBAN CONSUM	NORTH CENTRAL
284	284	1	1	ALL URBAN CONSUM	SOUTH
284	284	90	90	ALL URBAN CONSUM	SOUTH
92	92	7	7	ALL URBAN CONSUM	SOUTH
44	44	1	1	ALL URBAN CONSUM	SOUTH
284	284	1	1	ALL URBAN CONSUM	WEST
284	284	90	90	ALL URBAN CONSUM	WEST
92	92	7	7	ALL URBAN CONSUM	WEST
44	44	1	1	ALL URBAN CONSUM	WEST

Partial Listing of the OUTCONT= Data Set

name	select	type	length	varnum	label	format	format
ASL5	1	1	5	.	SERVICES LESS MEDICAL CARE	0	0
A0L5	1	1	5	.	ALL ITEMS LESS MEDICAL CARE	0	0
A5	1	1	5	.	MEDICAL CARE	0	0
A51	1	1	5	.	MEDICAL CARE COMMODITIES	0	0
A512	1	1	5	.	MEDICAL CARE SERVICES	0	0

The following statements make use of this information to extract the data for A512 and descriptive information on cross sections containing A512:

```
proc format;
   value $areafmt '0100' = 'Northeast Region'
                  '0200' = 'North Central Region'
                  '0300' = 'Southern Region'
                  '0400' = 'Western Region';
run;

filename datafile 'host-specific-file-name' <host-options>;
proc datasource filetype=blscpi interval=month
                out=medical outall=medinfo;
   where survey='CU' and area in ( '0100','0200','0300','0400' );
   keep a512;
   range  from 1980:5;
   format area $areafmt.;
   rename a512=medcare;
run;

title1 'Information on Medical Care Service';
proc print data=medinfo;
run;
```

Output 10.2.2. Printout of the OUTALL= Data Set

```
                      Information on Medical Care Service

                                   b                     b                s
                                   a        b            y                e
   s    s                          s        a       l    s                l
   u    e                          p        s       e    e                e
   r    a   a                      t        e       n    l     n      k   c   t
O  v    s   r                      y        p       g    e     a      e   t   y
b  e    o   e                      p        e       t    c     m      p   e   p
s  y    n   a                      e        r       h    t     e      t   d   e

1  CU   U   Northeast Region       S   1982-84=100  5    1   MEDCAR   1   1   1
2  CU   U   North Central Region   S   1982-84=100  5    1   MEDCAR   1   1   1
3  CU   U   Southern Region        S   1982-84=100  5    1   MEDCAR   1   1   1
4  CU   U   Western Region         S   1982-84=100  5    1   MEDCAR   1   1   1

                                                                     e
                                      f   f         s                n
   v     b                        f   o   o         t                d
   a     l              l         o   r   r         _                _    n
   r     k              a         r   m   m         d                d    t     n
O  n     n              b         m   a   a         a                a    i     o
b  u     u              e         a   t   t         t                t    m     b
s  m     m              l         t   l   d         e                e    e     s

1  7   3479   MEDICAL CARE SERVICES   0   0   DEC1977   JUL1990   152   152
2  7   3578   MEDICAL CARE SERVICES   0   0   DEC1977   JUL1990   152   152
3  7   3677   MEDICAL CARE SERVICES   0   0   DEC1977   JUL1990   152   152
4  7   3776   MEDICAL CARE SERVICES   0   0   DEC1977   JUL1990   152   152

    n            s            a
    i            u            r
    n            r            e                   s
    r            t            a                   _          u
    a            i            n                   c          n    n
O   n            t            a                   o          i    d
b   g            l            m                   d          t    e
s   e            e            e                   e          s    c

1  123    ALL URBAN CONSUM    NORTHEAST        CUUR0100SA512         1
2  123    ALL URBAN CONSUM    NORTH CENTRAL    CUUR0200SA512         1
3  123    ALL URBAN CONSUM    SOUTH            CUUR0300SA512         1
4  123    ALL URBAN CONSUM    WEST             CUUR0400SA512         1
```

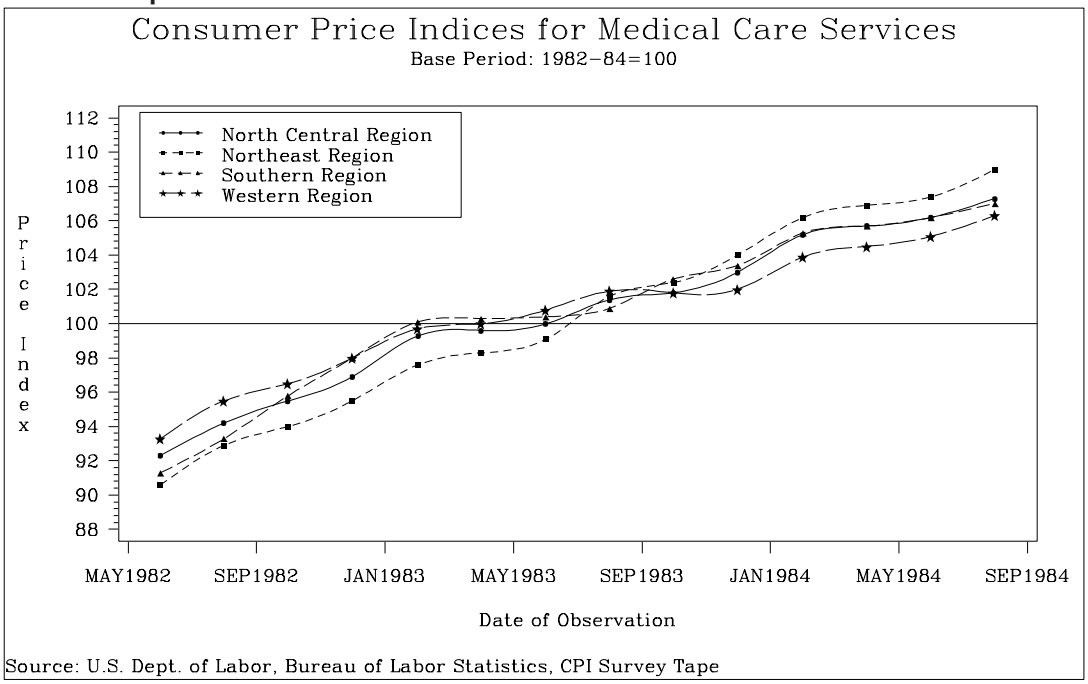
Note that only the cross sections with BASEPER='1982-84=100' are listed in the OUTALL= data set (see Output 10.2.2). This is because only those cross sections contain data for MEDCARE.

The OUTALL= data set indicates that data values are stored with one decimal place (see the NDEC variable). Therefore, they need to be rescaled, as follows:

```
data medical;
   set medical;
   medcare = medcare * 0.1;
run;
```

The variation of MEDCARE against DATE with respect to different geographic regions can be demonstrated graphically, as follows:

Output 10.2.3. Plot of Time Series in the OUT= Data Set for FILETYPE=BLSCPI



This example illustrates the following features:

- Descriptive information needed to write KEEP and WHERE statements can be obtained with an initial run of the DATASOURCE procedure.
- The OUTCONT= and OUTALL= data sets may contain information on how data values are stored, such as the precision, the units, and so on.
- The OUTCONT= and OUTALL= data sets report the new series names assigned by the RENAME statement, not the old names (see the NAME variable in Output 10.2.2).
- You can use PROC FORMAT to define formats for series or BY variables to enhance your output. Note that PROC DATASOURCE associated a permanent format, $AREAFMT., with the BY variable AREA. As a result, the formatted values are displayed in the printout of the OUTALL=MEDINFO data set (see Output 10.2.2) and in the legend created by PROC GPLOT.
- The base period for all the geographical areas is the same (BASEPER='1982-84=100') as indicated by the intersections of plots with the horizontal reference line drawn at 100. This makes comparisons meaningful.

Example 10.3. BLS State and Area, Employment, Hours and Earnings Surveys

This example illustrates how to extract specific series from a State and Area, Employment, Hours and Earnings Survey. The series to be extracted is total employment in manufacturing industries with respect to states as of March, 1990.

The State and Area, Employment, Hours and Earnings survey designates the totals for manufacturing industries by DIVISION='3', INDUSTRY='0000', and DETAIL='1'. Also, statewide figures are denoted by AREA='0000'.

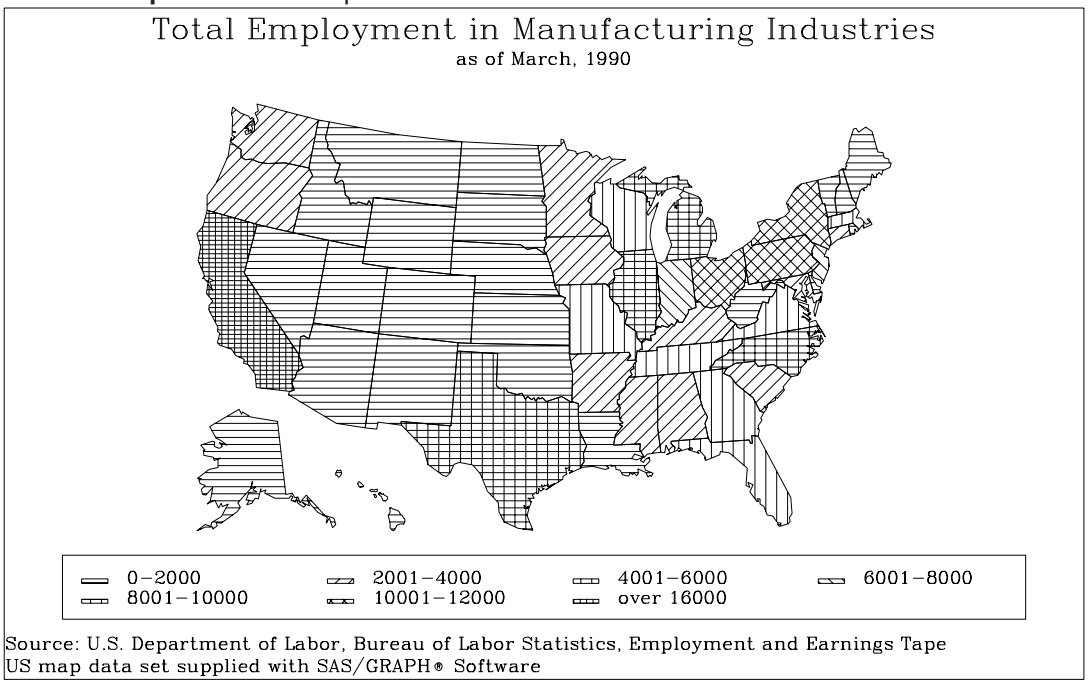
The data type code for total employment is reported to be 1. Therefore, the series name for this variable is SA1, since series names are constructed by adding an SA prefix to the data type codes given by BLS.

The following statements extract statewide figures for total employment (SA1) in manufacturing industries for March, 1990:

```
filename datafile 'host-specific-file-name' <host-options>;
proc datasource filetype=blseesa out=totemp;
   where division='3' and industry='0000' and detail='1' and
         area='0000';
   keep sa1;
   range from 1990:3 to 1990:3;
   rename sa1=totemp;
run;
```

Variations of women workers in manufacturing industries with respect to states can best be demonstrated on a map of the United States, as shown in Output 10.3.1.

Output 10.3.1. Map of the Series in the OUT= Data Set for FILETYPE=BLSEESA



Note the following for the preceding example:

- The INFILE= option is omitted, since the fileref assigned to the BLSEESA file is the default value DATAFILE.
- When the FROM and TO values in the RANGE statement are the same, only one observation for each cross section is extracted. This observation cor-

responds to a monthly data point since the INTERVAL= option defaults to MONTH.

Example 10.4. DRI/McGraw-Hill Tape Format CITIBASE Files

This example illustrates how to extract daily series from a sample CITIBASE file. Also, it shows how the OUTSELECT= option affects the contents of the auxiliary data sets.

The daily series contained in the sample data file CITIDEMO are listed by the following statements:

```
proc datasource filetype=citibase infile=citidemo interval=weekday
                outall=citiall outby=citikey;
run;

title1 'Summary Information on Daily Data for CITIDEMO File';
proc print data=citikey noobs;
run;

title1 'Daily Series Available in CITIDEMO File';
proc print data=citiall( drop=label );
run;
```

Output 10.4.1. Printout of the OUTBY= and OUTALL= Data Sets

Summary Information on Daily Data for CITIDEMO File

OBS	ST_DATE	END_DATE	NTIME	NOBS	NSERIES	NSELECT
1	01JAN1988	14MAR1991	835	835	10	10

Daily Series Available in CITIDEMO File 2

Obs	NAME	SELECTED	TYPE	LENGTH	VARNUM	BLKNUM
1	DSIUSNYDJCM	1	1	5	.	42
2	DSIUSNYSECM	1	1	5	.	43
3	DSIUSWIL	1	1	5	.	44
4	DFXWCAN	1	1	5	.	45
5	DFXWUK90	1	1	5	.	46
6	DSIUKAS	1	1	5	.	47
7	DSIJPND	1	1	5	.	48
8	DCP05	1	1	5	.	49
9	DCD1M	1	1	5	.	50
10	DTBD3M	1	1	5	.	51

Obs	LABEL	FORMAT
1	STOCK MKT INDEX:NY DOW JONES COMPOSITE, (WSJ)	
2	STOCK MKT INDEX:NYSE COMPOSITE, (WSJ)	
3	STOCK MKT INDEX:WILSHIRE 500, (WSJ)	
4	FOREIGN EXCH RATE WSJ:CANADA,CANADIAN $/U.S. $,NSA	
5	FOREIGN EXCH RATE WSJ:U.K.,CENTS/POUND(90 DAY FORWARD),NSA	
6	STOCK MKT INDEX:U.K. - ALL SHARES	
7	STOCK MKT INDEX:JAPAN - NIKKEI-DOW	
8	INT.RATE:5-DAY COMM.PAPER, SHORT TERM YIELD	
9	INT.RATE:1MO CERTIFICATES OF DEPOSIT, SHORT TERM YIELD (FBR H.15)	
10	INT.RATE:3MO T-BILL, DISCOUNT YIELD (FRB H.15)	

Obs	FORMATL	FORMATD	ST_DATE	END_DATE	NTIME	NOBS	ATTRIBUT	NDEC
1	0	0	04JAN1988	14MAR1991	834	834	1	2
2	0	0	04JAN1988	14MAR1991	834	834	1	2
3	0	0	04JAN1988	14MAR1991	834	834	1	2
4	0	0	01JAN1988	14MAR1991	835	835	1	4
5	0	0	01JAN1988	14MAR1991	835	835	1	2
6	0	0	01JAN1988	14MAR1991	835	835	1	2
7	0	0	01JAN1988	14MAR1991	835	835	1	2
8	0	0	04JAN1988	24FEB1989	300	300	2	2
9	0	0	04JAN1988	08MAR1991	830	830	1	2
10	0	0	04JAN1988	08MAR1991	830	830	1	2

Note the following from Output 10.4.1:

- The OUTALL= data set reports the time ranges of variables.
- There are ten observations in the OUTALL= data set, the same number as reported by NSERIES and NSELECT variables in the OUTBY= data set.
- The VARNUM variable contains all MISSING values, since no OUT= data set is created.

The next step is to demonstrate how the OUTSELECT= option affects the contents of the OUTBY= and OUTALL= data sets when a KEEP statement is present. First, set the OUTSELECT= option to OFF.

```
proc datasource filetype=citibase infile=citidemo interval=weekday
                outall=alloff outby=keyoff outselect=off;
   keep dsiusnysecm dc:;
run;
```

```
title1 'Summary Information on Daily Data for CITIDEMO File';
proc print data=keyoff;
run;

title1 'Daily Series Available in CITIDEMO File';
proc print data=alloff( keep=name kept selected st_date
                            end_date ntime nobs );
run;
```

Output 10.4.2. Printout of the OUTBY= and OUTALL= Data Sets with OUTSELECT=OFF

Summary Information on Daily Data for CITIDEMO File

OBS	ST_DATE	END_DATE	NTIME	NOBS	NSERIES	NSELECT
1	01JAN1988	14MAR1991	835	834	10	3

Daily Series Available in CITIDEMO File

Obs	NAME	KEPT	SELECTED	ST_DATE	END_DATE	NTIME	NOBS
1	DSIUSNYDJCM	0	0	04JAN1988	14MAR1991	834	834
2	DSIUSNYSECM	1	1	04JAN1988	14MAR1991	834	834
3	DSIUSWIL	0	0	04JAN1988	14MAR1991	834	834
4	DFXWCAN	0	0	01JAN1988	14MAR1991	835	835
5	DFXWUK90	0	0	01JAN1988	14MAR1991	835	835
6	DSIUKAS	0	0	01JAN1988	14MAR1991	835	835
7	DSIJPND	0	0	01JAN1988	14MAR1991	835	835
8	DCP05	1	1	04JAN1988	24FEB1989	300	300
9	DCD1M	1	1	04JAN1988	08MAR1991	830	830
10	DTBD3M	0	0	04JAN1988	08MAR1991	830	830

Then, set the OUTSELECT= option ON.

```
proc datasource filetype=citibase infile=citidemo interval=weekday
                outall=allon outby=keyon outselect=on;
   keep dsiusnysecm dc:;
run;

title1 'Summary Information on Daily Data for CITIDEMO File';
proc print data=keyon;
run;

title1 'Daily Series Available in CITIDEMO File';
proc print data=allon( keep=name kept selected st_date
                           end_date ntime nobs );
run;
```

Output 10.4.3. Printout of the OUTBY= and OUTALL= Data Sets with OUTSELECT=ON

Summary Information on Daily Data for CITIDEMO File

OBS	ST_DATE	END_DATE	NTIME	NOBS	NSERIES	NSELECT
1	04JAN1988	14MAR1991	834	834	10	3

Daily Series Available in CITIDEMO File

Obs	NAME	KEPT	SELECTED	ST_DATE	END_DATE	NTIME	NOBS
1	DSIUSNYSECM	1	1	04JAN1988	14MAR1991	834	834
2	DCP05	1	1	04JAN1988	24FEB1989	300	300
3	DCD1M	1	1	04JAN1988	08MAR1991	830	830

Comparison of Output 10.4.2 and Output 10.4.3 reveals the following:

- The OUTALL= data set contains ten (NSERIES) observations when OUTSELECT=OFF, and three (NSELECT) observations when OUTSELECT=ON.
- The observations in OUTALL=ALLON are those for which SELECTED=1 in OUTALL=ALLOFF.
- The time ranges in the OUTBY= data set are computed over all the variables (selected or not) for OUTSELECT=OFF, resulting in ST_DATE='01JAN88'd and END_DATE='14MAR91'd; and over only the selected variables for OUTSELECT=ON, resulting in ST_DATE='04JAN88'd and END_DATE='14MAR91'd. This corresponds to computing time ranges over all the series reported in the OUTALL= data set.
- The variable NTIME is the number of time periods between ST_DATE and END_DATE, while NOBS is the number of observations the OUT= data set is to contain. Thus, NTIME is different depending on whether the OUTSELECT= option is set to ON or OFF, while NOBS stays the same.

Also the use of the KEEP statement in the last two examples illustrates the use of an additional variable, KEPT, in the OUTALL= data sets of Output 10.4.2 and Output 10.4.3. KEPT, which reports the outcome of the KEEP statement, is only added to the OUTALL= data set when there is KEEP statement, as shown in Output 10.4.1.

Adding the RANGE statement to the last example generates the data sets in Output 10.4.4:

```
proc datasource filetype=citibase infile=citidemo interval=weekday
                outby=keyrange out=citiday outselect=on;
   keep dsiusnysecm dc:;
   range to '12jan88'd;
run;
```

```
title1 'Summary Information

title1 'Daily Data in CITIDEMO File';
proc print data=citiday;
run;
```

Output 10.4.4. Printout of the OUT=CITIDAY Data Set for FILETYPE=CITIBASE

Daily Series Available in CITIDEMO File

OBS	ST_DATE	END_DATE	NTIME	NOBS	NINRANGE	NSERIES	NSELECT
1	04JAN1988	14MAR1991	834	834	7	10	3

Daily Data in CITIDEMO File

Obs	DATE	DSIUSNYSECM	DCP05	DCD1M
1	04JAN1988	142.900	6.81000	6.89000
2	05JAN1988	144.540	6.84000	6.85000
3	06JAN1988	144.820	6.79000	6.87000
4	07JAN1988	145.890	6.77000	6.88000
5	08JAN1988	137.030	6.73000	6.88000
6	11JAN1988	138.810	6.81000	6.89000
7	12JAN1988	137.740	6.73000	6.83000

The OUTBY= data set in this last example contains an additional variable NINRANGE. This variable is added since there is a RANGE statement. Its value, 7, is the number of observations in the OUT= data set. In this case, NOBS gives the number of observations the OUT= data set would contain if there were not a RANGE statement.

Note that the OUT= data set does not contain data for 09JAN1988 and 10JAN1988. This is because the WEEKDAY interval skips over weekends.

Example 10.5. DRI Data Delivery Service Database

This example demonstrates the DRIDDS filetype for the daily Federal Reserve Series fxrates_dds. Use VALIDVARNAME=ANY on your SAS options statement to allow special characters such as @, $, and % to be allowed in the series name. Note the use of long variable names in the OUT= data set and long labels in the OUTCONT= data set.

The following statements extract the daily series starting from January 1,1997:

```
filename datafile 'host-specific-file-name' <host-options>;
proc format;
   value distekfm 0 = 'Unspecified'
                  2 = 'Linear'
                  4 = 'Triag'
                  6 = 'Polynomial'
                  8 = 'Even'
```

```
                    10 = 'Step'
                    12 = 'Stocklast'
                    14 = 'LinearUnadjusted'
                    16 = 'PolyUnadjusted'
                    18 = 'StockWithNAS'
                    99 = 'None'
                   255 = 'None';


   value convtkfm 0 = 'Unspecified'
                   1 = 'Average'
                   3 = 'AverageX'
                   5 = 'Sum'
                   7 = 'SumAnn'
                   9 = 'StockEnd'
                  11 = 'StockBegin'
                  13 = 'AvgNP'
                  15 = 'MaxNP'
                  17 = 'MinNP'
                  19 = 'StockEndNP'
                  21 = 'StockBeginNP'
                  23 = 'Max'
                  25 = 'Min'
                  27 = 'AvgXNP'
                  29 = 'SumNP'
                  31 = 'SumAnnNP'
                  99 = 'None'
                 255 = 'None';

 /*-------------------------------------------------------------*
  *                process daily series                         *
  *-------------------------------------------------------------*/
title3 'Reading DAILY Federal Reserve Series with fxrates_.dds';
proc datasource filetype=dridds
                infile=datafile
                interval=day
                out=fixr
                outcont=fixrcnt
                outall=fixrall;

   range from '01jan97'd to '31dec99'd;
   format disttek distekfm.;
   format convtek convtkfm.;
run;
```

Example 10.6. PC Diskette Format CITIBASE Database

This example uses a diskette format CITIBASE database (FILETYPE=CITIDISK) to extract annual population estimates for females and males with respect to various age groups since 1980.

Population estimate series for females with five-year age intervals are given by PANF1 through PANF16, where PANF1 is for females under 5 years of age, PANF2

is for females between 5 and 9 years of age, and so on. Similarly, PANM1 through PANM16 gives population estimates for males with five-year age intervals.

The following statements extract the required population estimates series:

```
filename keyfile 'host-specific-key-file-name' <host-options>;
filename indfile 'host-specific-ind-file-name' <host-options>;
filename dbfile  'host-specific-db-file-name'  <host-options>;
proc datasource filetype=citidisk  infile=( keyfile indfile dbfile )
                out=popest outall=popinfo;
   keep panf1-panf16 panm1-panm16;
   range from 1980;
run;
```

This example demonstrates the following:

- The INFILE= options lists the filerefs of the key, index, and database files, in that order.
- The INTERVAL= option is omitted since the default interval for CITIDISK type files is YEAR.

Example 10.7. Quarterly COMPUSTAT Data Files

This example shows how to extract data from a 48-quarter Compustat Database File. For COMPUSTAT data files, the series variable names are constructed by concatenating the name of the data array DATA and the column number containing the required information. For example, for quarterly files the common stock data is in column 56. Therefore, the variable name for this series is DATA56. Similarly, the series variable names for quarterly footnotes are constructed by adding the column number to the array name, QFTNT. For example, the variable name for common stock footnotes is QFTNT14 since the 14th column of the QFTNT array contains this information.

The following example extracts common stock series (DATA56) and its footnote (QFTNT14) for Computer Programming Service Companies (DNUM=7371) and Prepackaged Software Companies (DNUM=7370) whose stocks are traded over-the-counter and not in the S&P 500 Index (ZLIST=06) and whose data reside in the over-the-counter file (FILE=06).

```
filename compstat 'host-specific-Compustat-file-name' <host-options>;
proc datasource filetype=cs48qibm infile=compstat
                out=stocks outby=company;
   keep data56 qftnt14;
   rename data56=comstock  qftnt14=ftcomstk;
   label  data56='Common Stock'
          qftnt14='Footnote for Common Stock';
   where dnum in (7370,7371) and zlist=06 and file=06;
run;

/*- add company name to the out= data set    */
```

```
data stocks;
   merge stocks company( keep=dnum cnum cic coname );
   by dnum cnum cic;
run;

title1 'Common Stocks for Software Companies for 1990';
proc print data=stocks noobs;
   where date between '01jan90'd and '31dec90'd;
run;
```

The Output 10.7.1 contains a partial listing of the STOCKS data set.

Output 10.7.1. Partial Listing of the OUT=STOCKS Data Set

Common Stocks for Software Companies for 1990

DNUM	CNUM	CIC	FILE	EIN	STK	SMBL	ZLIST	XREL	FINC	SINC	state
7370	027352	103	6	54-0856778	0	AMSY	6	0	0	10	51
7370	027352	103	6	54-0856778	0	AMSY	6	0	0	10	51
7370	027352	103	6	54-0856778	0	AMSY	6	0	0	10	51
7370	027352	103	6	54-0856778	0	AMSY	6	0	0	10	51
7370	553412	107	6	73-1064024	0	MPSG	6	0	0	10	40
7370	553412	107	6	73-1064024	0	MPSG	6	0	0	10	40
7370	553412	107	6	73-1064024	0	MPSG	6	0	0	10	40
7370	553412	107	6	73-1064024	0	MPSG	6	0	0	10	40
7371	032681	108	6	41-0905408	0	ANLY	6	0	0	27	27
7371	032681	108	6	41-0905408	0	ANLY	6	0	0	27	27
7371	032681	108	6	41-0905408	0	ANLY	6	0	0	27	27
7371	032681	108	6	41-0905408	0	ANLY	6	0	0	27	27
7371	458816	105	6	04-2448936	0	IMET	6	0	0	25	25
7371	458816	105	6	04-2448936	0	IMET	6	0	0	25	25
7371	458816	105	6	04-2448936	0	IMET	6	0	0	25	25
7371	458816	105	6	04-2448936	0	IMET	6	0	0	25	25
7371	834021	107	6	04-2453033	0	SOFT	6	0	0	25	25
7371	834021	107	6	04-2453033	0	SOFT	6	0	0	25	25
7371	834021	107	6	04-2453033	0	SOFT	6	0	0	25	25
7371	834021	107	6	04-2453033	0	SOFT	6	0	0	25	25
7371	872885	108	6	13-2635899	0	TSRI	6	0	0	10	36
7371	872885	108	6	13-2635899	0	TSRI	6	0	0	10	36
7371	872885	108	6	13-2635899	0	TSRI	6	0	0	10	36
7371	872885	108	6	13-2635899	0	TSRI	6	0	0	10	36
7371	878351	105	6	41-0918564	0	TECN	6	0	0	27	27
7371	878351	105	6	41-0918564	0	TECN	6	0	0	27	27
7371	878351	105	6	41-0918564	0	TECN	6	0	0	27	27
7371	878351	105	6	41-0918564	0	TECN	6	0	0	27	27

county	date	comstock	ftcomstk	CONAME
13	1990:1	0.11500		AMERICAN MANAGEMENT SYSTEMS
13	1990:2	0.11600		AMERICAN MANAGEMENT SYSTEMS
13	1990:3	0.12200		AMERICAN MANAGEMENT SYSTEMS
13	1990:4	0.11700		AMERICAN MANAGEMENT SYSTEMS
143	1990:1	0.42400		MPSI SYSTEMS INC
143	1990:2	0.42400		MPSI SYSTEMS INC
143	1990:3	0.42400		MPSI SYSTEMS INC
143	1990:4	0.42300		MPSI SYSTEMS INC
53	1990:1	.		ANALYSTS INTERNATIONAL CORP
53	1990:2	.		ANALYSTS INTERNATIONAL CORP
53	1990:3	.		ANALYSTS INTERNATIONAL CORP
53	1990:4	0.46000		ANALYSTS INTERNATIONAL CORP
17	1990:1	0.03600		INTERMETRICS INC
17	1990:2	0.03600		INTERMETRICS INC
17	1990:3	0.03600		INTERMETRICS INC
17	1990:4	.		INTERMETRICS INC
17	1990:1	0.38700		SOFTECH INC
17	1990:2	0.38700		SOFTECH INC
17	1990:3	.		SOFTECH INC
17	1990:4	.		SOFTECH INC
103	1990:1	0.02500		TSR INC
103	1990:2	0.02500		TSR INC
103	1990:3	.		TSR INC
103	1990:4	.		TSR INC
53	1990:1	0.21500		TECHNALYSIS CORP
53	1990:2	0.21600		TECHNALYSIS CORP
53	1990:3	0.21600		TECHNALYSIS CORP
53	1990:4	0.21600		TECHNALYSIS CORP

Note that quarterly Compustat data are also available in Universal Character format. If you have this type of file instead of IBM 360/370 General format, use the FILE-TYPE=CS48QUC option instead.

Example 10.8. Annual COMPUSTAT Data Files

This example shows how to extract a subset of cross sections when the required cross sections are listed in an external file. In the case of a COMPUSTAT file, the required cross sections are a list of companies. For example, you may want to extract annual data for a list of companies whose industry classification codes (DNUM), CUSIP issuer codes (CNUM), and CUSIP issue number and check digits (CIC) are given in an external file, COMPLIST, as follows:

```
2640   346377   104
3714   017634   106
5812   171583   107
6025   446150   104
8051   087851   101
```

When the required companies are listed in an external file, you can either use the SAS macro processor to construct your WHERE statement expression or restructure your data file and include it after the WHERE key word.

The following steps use the first approach to construct the WHERE statement expression in the macro variable WHEXPR:

```
filename compfile 'host-specific-file-name' <host-options>;
%macro whstmt( fileref );
   %global whexpr;
   data _null_;
      infile &fileref end=last;
      length cnum $ 6;
      input  dnum cnum cic;
      call symput( 'dnum'||left(_n_), left(dnum) );
      call symput( 'cnum'||left(_n_), cnum );
      call symput( 'cic' ||left(_n_), left(cic) );
      if last then call symput( 'n', left(_n_) );
   run;
   %do i = 1 %to &n;
      %let whexpr = &whexpr
       (DNUM=&&dnum&i and CNUM="&&cnum&i" and CIC=&&cic&i);
      %if &i ^= &n %then %let whexpr = &whexpr or;
      %end;
   %mend whstmt;
%whstmt( compfile );
filename compustat 'host-specific-Compustat-file-name' <host-options>;
proc datasource filetype=csaibm infile=compstat
               outby=company  out=dataset;
   where &whexpr;
run;
```

The same result can also be obtained by creating an external file, WHEXPR, from the COMPFILE and including it after the WHERE key word, as shown in the following statements:

```
filename whexpr 'host-specific-WHEXPR-file-name' <host-options>;
data _null_;
   infile compfile end=last; file   whexpr;
   length cnum $ 6;
   input  dnum cnum cic;
   put "( "  dnum=  "and CNUM='"  cnum $6.  "' and "  cic= ")"  @;
   if not last then put ' or'; else put ';' ;
run;

filename compstat 'host-specific-Compustat-file-name' <host-options>;
proc datasource filetype=csaibm infile=compustat
                outby=company  out=dataset;
   where %inc 'host-specific-WHEXPR-file-name';
run;

title1 'Information on Selected Companies';
proc print data=company;
run;
```

The Output 10.8.1 shows the OUTBY= data set created by the preceding statements. As you can see, the companies listed in the COMPLIST file are reported in this data set.

Output 10.8.1. Printout of the OUTBY= Data Set Listing Selected Companies

```
                       Information on Selected Companies




                                                                C
                              Z                          S      O
    D      C              F   L    S         X           T      U    F
O   N      N       C      I   I    M         R    S      A      N    I       E
b   U      U       I      L   S    B         E    T      T      T    N       I
s   M      M       C      E   T    L         L    K      E      Y    C       N

1  2640   346377   104    3   4   FOR        0    0     34     31    0   34-1046753
2  3714   017634   106    1   4   ALN        0    0     36    103    0   38-0290950
3  5812   171583   107   11   1   CHU     5812    0     48     29    0   74-1507270
4  6025   446150   104    3   6   HBAN       0    0     39     49    0   31-0724920
5  8051   087851   101   11   1   BEV     8050    0      6     37    0   95-4100309

   b            e
   y      s     n            n     n
   s      t     d            s     s
   e      _     _    n       e     e        I
   l      d     d    t   n   r     l        N
O  e      a     a    i   o   i     e    R   A
b  c      t     t    m   b   e     c    E   M
s  t      e     e    e   s   s     t    C   E

1  1    1968   1987   20   20   423   366   1   CONVRT,PAPRBRD PD,EX CONTAIN
2  1    1968   1987   20   20   423   366   1   MOTOR VEHICLE PART,ACCESSORY
3  1    1968   1987   20   20   423   366   1   EATING PLACES
4  1    1968   1987   20   20   423   366   1   NATL BANKS-FED RESERVE SYS
5  1    1968   1987   20   20   423   366   1   SKILLED NURSING CARE FAC



   C
   O                                 D       C                   F
   N                                 N       N       C     R     I
O  A                          D      U       U       I     E     L
b  M                          U      M       M       C     C     E
s  E                          P      2       2       2     2     2

1  FORMICA CORP               0    2640    346377    104   2      3
2  ALLEN GROUP                0    3714    017634    106   2      1
3  CHURCH'S FRIED CHICKEN INC 0    5812    171583    107   2     11
4  HUNTINGTON BANCSHARES      0    6025    446150    104   2      3
5  BEVERLY ENTERPRISES        0    8051    087851    101   2     11
```

Note that annual COMPUSTAT data are available in either IBM 360/370 General format or the Universal Character format. The first example expects an IBM 360/370 General format file since the FILETYPE= is set to CSAIBM, while the second example uses a Universal Character format file (FILETYPE=CSAUC).

Example 10.9. CRSP Daily NYSE/AMEX Combined Stocks

This example reads all the data on a three-volume daily NYSE/AMEX combined character data set. Assume that the following filerefs are assigned to the calendar/indices file and security files comprising this database:

Fileref	VOLSER	File Type
calfile	DXAA1	calendar/indices file on volume 1
secfile1	DXAA1	security file on volume 1
secfile2	DXAA2	security file on volume 2
secfile3	DXAA3	security file on volume 3

The data set CALDATA is created by the following statements to contain the calendar/indices file:

```
proc datasource filetype=crspdci infile=calfile out=caldata;
run;
```

Here the FILETYPE=CRSPDCI indicates that you are reading a character format (indicated by a C in the 6th position) daily (indicated by a D in the 5th position) calendar/indices file (indicated by an I in the 7th position).

The annual data in security files can be obtained by the following statements:

```
proc datasource filetype=crspdca
                infile=( secfile1 secfile2 secfile3 )
                out=annual;
run;
```

Similarly, the data sets to contain the daily security data (the OUT= data set) and the event data (the OUTEVENT= data set) are obtained by the following statements:

```
proc datasource filetype=crspdcs
                infile=( calfile secfile1 secfile2 secfile3 )
                out=periodic index outevent=events;
run;
```

Note that the FILETYPE= has an S at the 7th position, since you are reading the security files. Also, the INFILE= option first expects the fileref of the calendar/indices file since the dating variable (CALDT) is contained in that file. Following the fileref of calendar/indices file, you give the list of security files in the order you want to read them.

The Output 10.9.1 is generated by the following statements:

```
title1 'First 5 Observations in the Calendar/Indices File';
proc print data=caldata( obs=5 );
run;

title1 'Last 5 Observations in the Calendar/Indices File';
proc print data=caldata( firstobs=6659 ) noobs;
run;

title1 "Periodic Series for CUSIP='09523220'";
title2 "DATE >= '22dec88'd";
```

```
proc print data=periodic;
   where cusip='09523220' and date >= '22dec88'd;
run;

title1 "Events for CUSIP='09523220'";
proc print data=events;
   where cusip='09523220';
run;
```

Output 10.9.1. Partial Listing of the Output Data Sets

First 5 Observations in the Calendar/Indices File

Obs	date	VWRETD	VWRETX	EWRETD	EWRETX	TOTVAL
1	02JUL1962	-99.0000	-99.0000	-99.0000	-99.0000	319043897
2	03JUL1962	0.0113	0.0112	0.0131	0.0130	322929231
3	05JUL1962	0.0060	0.0059	0.0069	0.0068	324750979
4	06JUL1962	-0.0107	-0.0107	-0.0064	-0.0064	321302641
5	09JUL1962	0.0067	0.0067	0.0018	0.0018	323221296

Obs	TOTCNT	USDVAL	USDCNT	SPINDX	SPRTRN
1	2036	0	0	55.86	-99.0000
2	2040	319043897	2036	56.49	0.0113
3	2031	322838977	2031	56.81	0.0057
4	2031	324699079	2022	56.17	-0.0113
5	2029	320935790	2019	56.55	0.0068

Last 5 Observations in the Calendar/Indices File

date	VWRETD	VWRETX	EWRETD	EWRETX	TOTVAL
23DEC1988	0.0042154	0.0028936	0.005104	0.003588	2367541510
27DEC1988	-.0029128	-.0029624	-0.001453	-0.001585	2360680550
28DEC1988	0.0015624	0.0015249	0.001575	0.001484	2364369540
29DEC1988	0.0067816	0.0066433	0.005578	0.005469	2379932980
30DEC1988	-.0027338	-.0029144	0.010736	0.010572	2362374030

TOTCNT	USDVAL	USDCNT	SPINDX	SPRTRN
2563	2360655540	2561	277.87	0.0036118
2565	2367496320	2562	276.83	-.0037429
2568	2360668370	2564	277.08	0.0009031
2565	2364169480	2563	279.40	0.0083724
2567	2379932980	2565	277.72	-.0060126

```
                    Periodic Series for CUSIP='09523220'
                             DATE >= '22dec88'd

             P   C I     H                                                S B
     C       E   O S H   S             B      A                           X X
     U       R   M S E   I         D   I      S                           X R
O    S       M   P U X   C         A   D      K      P     V       R      R R
b    I       N   N N C   C         T   L      H      R     O       E      E E
s    P       O   O O D   D         E   O      I      C     L       T      T T

 3 09523220 75285 0 0 1 7361 22DEC1988 15.00 15.375 15.375 54300  0.016529 . .
 4 09523220 75285 0 0 1 7361 23DEC1988 15.50 15.750 15.625 17700  0.016260 . .
 5 09523220 75285 0 0 1 7361 27DEC1988 15.50 15.750 15.625 10600  0.000000 . .
 6 09523220 75285 0 0 1 7361 28DEC1988 15.50 15.500 15.500 10600 -0.008000 . .
 7 09523220 75285 0 0 1 7361 29DEC1988 15.25 15.500 15.375  7000 -0.008065 . .
 8 09523220 75285 0 0 1 7361 30DEC1988 15.00 15.250 15.000 13700 -0.024390 . .
```

```
                              Events for CUSIP='09523220'

               P     C   I         H                                N        T
     C         E     O   S    H    S      E                        C        I
     U         R     M   S    E    I      V              D         U        C
O    S         M     P   U    X    C      E              A         S        K
b    I         N     N   N    C    C      N              T         I        E
s    P         O     O   O    D    D      T              E         P        R

1  09523220   75285   0   0   1   7361   NAMES    03MAY1988   09523220   BAW
2  09523220   75285   0   0   1   7361   DIST     18JUL1988
3  09523220   75285   0   0   1   7361   SHARES   03MAY1988
4  09523220   75285   0   0   1   7361   SHARES   30SEP1988
5  09523220   75285   0   0   1   7361   SHARES   30DEC1988
6  09523220   75285   0   0   1   7361   DELIST   30DEC1988

         C         S     E         D      D        F        D        R
         O         H  S  X    S     I      I     F  A        C        C
         M         R  H  C    I     S      V     A  C        L        R
O        N         C  R  H    C     T      A     C  S        R        D
b        A         L  C  C    C     C      M     P  H        D        D
s        M         S  D  D    D     D      T     R  R        T        T

1  BLUE ARROW PLC     3  1  7361     .   .        .  .        .        .
2                     .  .     .  1212  0.13376  0  0  13JUL88  22JUL88
3                     .  .     .     .   .        .  .        .        .
4                     .  .     .     .   .        .  .        .        .
5                     .  .     .     .   .        .  .        .        .
6                     .  .     .     .   .        .  .        .        .

               S     S    D    N         N                          T   N       N
         P     H     H    L    W         E   D   D   D   D   D   R   M   M   S
         A     R     R    S    P         X   L   L   L   L   L   T   S   M   D
O        Y     O     F    T    E         T   B   A   P   V   R   S   I   C   I
b        D     U     L    C    R         D   I   S   R   O   E   C   N   N   N
s        T     T     G    D    M         T   D   K   C   L   T   D   D   T   X

1        .        .   .     .   .         .   .   .   .   .   .   .   .   .   .
2  26AUG88        .   .     .   .         .   .   .   .   .   .   .   .   .   .
3        .    72757   0     .   .         .   .   .   .   .   .   .   .   .   .
4        .   706842   0     .   .         .   .   .   .   .   .   .   .   .   .
5        .   706842   0     .   .         .   .   .   .   .   .   .   .   .   .
6        .        .   .   100   0         .   .   .   0   .   A   .   .   .   .
```

This example illustrates the following points:

- When data span more than one physical volume, the filerefs of the security files residing on each volume must be given following the fileref of the calendar/indices file. The DATASOURCE procedure reads each of these files in the order they are specified. Therefore, you can request that all three volumes be mounted to the same tape drive, if you choose to do so.
- The INDEX option in the second PROC DATASOURCE run creates an index file for the OUT=PERIODIC data set. This index file provides random access to the OUT= data set and may increase the efficiency of the subsequent PROC and DATA steps that use BY and WHERE statements. The index variables are CUSIP, CRSP permanent number (PERMNO), NASDAQ company number (COMPNO), NASDAQ issue number (ISSUNO), header exchange code (HEXCD) and header SIC code (HSICCD). Each one of these variables forms a different key, that is, a single index. If you want to form keys from a combination of variables (composite indexes) or use some other variables as indexes, you should use the INDEX= data set option for the OUT= data set.
- The OUTEVENT=EVENTS data set is sparse. In fact, for each EVENT type, a unique set of event variables are defined. For example, for EVENT='SHARES', only the variables SHROUT and SHRFLG are defined, and they have missing values for all other EVENT types. Pictorially, this structure is similar to the data set shown in Figure 10.8. Because of this sparse representation, you should create the OUTEVENT= data set only when you need a subset of securities and events.

By default, the OUT= data set contains only the periodic data. However, you may also want to include the event-oriented data in the OUT= data set. This is accomplished by listing the event variables together with periodic variables in a KEEP statement. For example, if you want to extract the historical CUSIP (NCUSIP), number of shares outstanding (SHROUT), and dividend cash amount (DIVAMT) together with all the periodic series, use the following statements:

```
proc datasource filetype=crspdcs
                infile=( calfile secfile1 secfile2 secfile3 )
                out=both outevent=events;
   where cusip='09523220';
   keep  bidlo askhi prc vol ret sxret bxret ncusip shrout divamt;
run;

proc datasource filetype=crspdcs
                infile=( calfile secfile1 )
                out=both outevent=events;
   where cusip='09523220';
   keep  bidlo askhi prc vol ret sxret bxret ncusip shrout divamt;
run;

proc datasource filetype=crspdcs
                infile=( calfile secfile1 )
                out=both2 outevent=events2;
   where cusip='09523220';
   keep  bidlo askhi prc vol ret sxret bxret ncusip shrout divamt;
   keepevent ncusip shrflg;
run;
```

```
title1 "Printout of the First 4 Observations";
title2 "CUSIP = '09523220'";
proc print data=both noobs;
   var  cusip date vol ncusip divamt shrout;
   where cusip='09523220' and date <= '08may88'd;
run;

title1 "Printout of the Observations centered Around 18jul88";
title2 "CUSIP = '09523220'";
proc print data=both noobs;
   var  cusip date vol ncusip divamt shrout;
   where cusip='09523220' and
         date between '14jul88'd and '20jul88'd;
run;

title1 "Printout of the Observations centered Around 30sep88";
title2 "CUSIP = '09523220'";
proc print data=both noobs;
   var  cusip date vol ncusip divamt shrout;
   where cusip='09523220' and
         date between '28sep88'd and '04oct88'd;
run;
```

Output 10.9.2. Including Event Variables in the OUT= Data Set

Printout of the First 4 Observations
CUSIP = '09523220'

CUSIP	DATE	VOL	NCUSIP	DIVAMT	SHROUT
09523220	03MAY1988	296100	09523220	.	72757
09523220	04MAY1988	139200	09523220	.	72757
09523220	05MAY1988	9000	09523220	.	72757
09523220	06MAY1988	7900	09523220	.	72757

Printout of the Observations centered Around 18jul88
CUSIP = '09523220'

CUSIP	DATE	VOL	NCUSIP	DIVAMT	SHROUT
09523220	14JUL1988	62000	09523220	.	72757
09523220	15JUL1988	106800	09523220	.	72757
09523220	18JUL1988	32100	09523220	0.13376	72757
09523220	19JUL1988	8600	09523220	.	72757
09523220	20JUL1988	10700	09523220	.	72757

Printout of the Observations centered Around 30sep88
CUSIP = '09523220'

CUSIP	DATE	VOL	NCUSIP	DIVAMT	SHROUT
09523220	28SEP1988	33000	09523220	.	72757
09523220	29SEP1988	55200	09523220	.	72757
09523220	30SEP1988	40700	09523220	.	706842
09523220	03OCT1988	13400	09523220	.	706842
09523220	04OCT1988	110600	09523220	.	706842

Events referring to distributions and delistings have entries only in observations whose dates match the event dates. For example, DIVAMT has a value for only 18JUL88, as shown in the second printout in Output 10.9.2. The NAME and SHARES events refer to a date of change, therefore their values are expanded such that there is a value for each observation. For example, the date of NAMES record is 03MAY88, therefore NCUSIP has the same value from that date on. The SHROUT on the other hand changes its value twice, once on 03MAY88, the other time on 30SEP88. The third listing shows how the value of SHROUT remains constant at 72757 from 03MAY88 to 30SEP88, at which date it changes to 706842.

The events occurring on days other than the trading dates are not output to the OUT= data set.

The KEEP statement in the preceding example has no effect on the event variables output to the OUTEVENT= data set. If you want to extract only a subset of event variables, you need to use the KEEPEVENT statement. For example, the following code outputs only NCUSIP and SHROUT to the OUTEVENT= data set for CUSIP='09523220':

```
proc datasource filetype=crspdxc
                infile=( calfile secfile1 secfile2 secfile3 )
                outevent=subevts;
   where cusip='09523220';
   keepevent  ncusip shrout;
run;

proc datasource filetype=crspdxc
                infile=( calfile secfile1)
                outevent=subevts;
   where cusip='09523220';
   keepevent  ncusip shrout;
run;

title1 "NCUSIP and SHROUT for CUSIP='09523220'";
proc print data=subevts noobs;
run;
```

Output 10.9.3. Listing of the OUTEVENT= Data Set with a KEEPEVENT Statement

NCUSIP and SHROUT for CUSIP='09523220'

CUSIP	PERMNO	COMPNO	ISSUNO	HEXCD	HSICCD	EVENT	DATE	NCUSIP	SHROUT
09523220	75285	0	0	1	7361	NAMES	03MAY1988	09523220	.
09523220	75285	0	0	1	7361	SHARES	03MAY1988		72757
09523220	75285	0	0	1	7361	SHARES	30SEP1988		706842
09523220	75285	0	0	1	7361	SHARES	30DEC1988		706842

The OUTEVENT= data set in Output 10.9.3 is missing observations for which the EVENT variable is DIST or DELIST, since these event groups do not contain any selected events.

Example 10.10. CRSP 1995 CDROM Data Files

The normal character filetypes used for tape files may also be used for the CDROM character data. They are CRSPDCS, CRSPDCI, CRSPDCA for daily data and CRSPMCS, CRSPMCI, CRSPMCA for monthly data. It is necessary to use the LRECL=(130 401) option on the DATASOURCE statement when processing CDROM character data as shown in the first DATASOURCE run.

The CRSP 1995 UNIX (SUN)Binary data is readable by PROC DATASOURCE using the filetypes CRSPDUS, CRSPDUI, CRSPDUA for daily files and filetypes CRSPMUS, CRSPMUI, CRSPMUA for monthly files. Both IEEE Big Endian and IEEE Little Endian machines may use the same CDROM UNIX Binary filetypes. PROC DATASOURCE can not read the PC Binary Data from CDROM, but the UNIX (SUN) Binary may be used from the same CDROM instead, even on PC's. The second DATASOURCE run shows how to access the 1995 UNIX Binary data.

```
filename csec 'machar.dat' recfm=f lrecl=401;
filename ccal 'msix.dat' recfm=f lrecl=130;

/*-------------------------------------------------------------*
 *       create output data sets without any subsetting        *
 *       character data from MA CDROM                          *
 *-------------------------------------------------------------*/
/*- create calendar/indices output data sets using DATASOURCE  -*/
/*- statements                                                 -*/
proc datasource filetype=crspmcs
                infile=( ccal csec )
                lrecl=( 130 401 )
                interval=month
                outselect=off
                outcont=maccont outkey=mackey
                out=mac outevent=macevent;
   keep _all_;
   keepevent _all_;
run;

title3 'MA/CDROM Security File Outputs';
title4 'OUTKEY= Data Set';
proc print data=mackey; run;

title4 'OUTCONT= Data Set';
proc print data=maccont; run;

title4 'Listing of OUT= Data Set';
proc print data=mac; run;

title4 'Listing of OUTEVENT= Data Set';
proc print data=macevent; run ;


filename macal 'maucal95.data' lrecl=48;
filename masec 'mausub95.data' recfm=v lrecl=32760;

/*-------------------------------------------------------------*
 *       create output data sets without any subsetting        *
 *       UNIX (SUN) binary data from MA CDROM                  *
```

```
 *-------------------------------------------------------------*/
/*- create calendar/indices output data sets using DATASOURCE -*/
/*- statements                                                 -*/
proc datasource filetype=crspmus
               infile=( macal masec )
               interval=month
               outselect=off
               outcont=macont outkey=makey
               out=ma outevent=maevent;
   keep _all_;
   keepevent _all_;
run;

title3 'MA/CDROM Security File Outputs';
title4 'OUTKEY= Data Set';
proc print data=makey; run;

title4 'OUTCONT= Data Set';
proc print data=macont; run;

title4 'Listing of OUT= Data Set';
proc print data=ma; run;

title4 'Listing of OUTEVENT= Data Set';
proc print data=maevent; run ;
```

Example 10.11. CRSP ACCESS97 CDROM Data Files

This example demonstrates how to work with the CRSP ACCESS97 CDROM data files by first running the CRSP supplied *stk_dump_bin* utility, to create a UNIX (SUN) binary file. The UNIX binary file can then be processed by PROC DATASOURCE using the CRSPMUS filetype for monthly data or the CRSPDUS filetype for DAILY data.

The DATASOURCE procedure expects the data file to use IEEE big Endian byte ordering. The exact command that you need to use to convert the data depends on whether you extracted the big Endian or little Endian data off of the CD, and whether you are running on a host whose native binary representation is big or little Endian. Consult your *1997 CRSP ACCESS97 Stock File User's Guide, Appendix C* for details on the reverse/keep option for the byte-ordering flag. Assuming a Windows NT platform and daily data:

```
ind_dump_bin %crsp_dstk% filename1 460 1000080 1000081 1000502 reverse unix
stk_dump_bin %crsp_dstk% filename2 10 1 0 0 0 reverse unix permlist_filename
```

Once you have converted the ACCESS97 data into the unix binary format, you are ready to invoke PROC DATASOURCE:

```
filename calfile 'filename1';
filename secfile 'filename2' lrecl=36000;

proc datasource filetype=crspdus
   infile=( calfile secfile )
```

```
   interval=day
   outselect=off
   out=da outkey=dakey outcont=dacont outevent=daevent;
   keep _all_;
   keepevent _all_;
run;
```

The above example uses an LRECL to accomodate the size of the 1997 daily data. Subsequent years may need a larger lrecl.

Example 10.12. IMF Direction of Trade Statistics

This example illustrates how to extract data from a Direction of Trade Statistics (DOTS) data file. The DOTS data files contain only two series, EXPORTS and IMPORTS, for various sets of countries. The foreign trade figures between any two countries can be extracted by specifying their three-digit codes for COUNTRY and PARTNER BY variables. The following statements can then be used to extract quarterly EXPORTS and IMPORTS between the United States of America (COUNTRY='111') and Japan (PARTNER='158').

```
filename dotsfile 'host-specific-gfs-file-name' <host-options>;
proc datasource filetype=imfdotsp infile=dotsfile interval=qtr
                out=foreign outall=forngvar;
   where country='111' and partner='158';
run;
```

References

Bureau of Economic Analysis (1986), *The National Income and Product Accounts of the United States, 1929-82*, U.S. Dept of Commerce, Washington D.C.

Bureau of Economic Analysis (1987), *Index of Items Appearing in the National Income and Product Accounts Tables*, U.S. Dept of Commerce, Washington D.C.

Bureau of Economic Analysis (1991), *Survey of Current Business*, U.S. Dept of Commerce, Washington D.C.

Center for Research in Security Prices (1997), *CRSP Stock 1996 File Guide*, Chicago, IL.

Center for Research in Security Prices (1997), *CRSP Access97 Stock File User's Guide*, Chicago, IL.

Center for Research in Security Prices (1997), *CRSP Stock 1997 File Programmer's Guide*, Chicago, IL.

Citibank (1990), *CITIBASE Directory*, New York, NY.

Citibank (1991), *CITIBASE-Weekly*, New York, NY.

Citibank (1991), *CITIBASE-Daily*, New York, NY.

DRI/McGraw-Hill (1997), *DataLink*, Lexington, MA.

DRI/McGraw-Hill Data Search and Retrieval for Windows (1996), *DRIPRO User's Guide*, Lexington, MA.

FAME Information Services (1995), *User's Guide to FAME*, Ann Arbor, Michigan

International Monetary Fund (1984), *IMF Documentation on Computer Tape Subscription*, Washington, D.C.

Organization For Economic Cooperation and Development (1992) *Annual National Accounts: Volume I. Main Aggregates Content Documentation for Magnetic Tape Subscription*, Paris, France.

Organization For Economic Cooperation and Development (1992) *Annual National Accounts: Volume II. Detailed Tables Technical Documentation for Magnetic Tape Subscription*, Paris, France.

Organization For Economic Cooperation and Development (1992) *Main Economic Indicators Database Note*, Paris, France.

Organization For Economic Cooperation and Development (1992) *Main Economic Indicators Inventory*, Paris, France.

Organization For Economic Cooperation and Development (1992) *Main Economic Indicators OECD Statistics on Magnetic Tape Document*, Paris, France.

Organization For Economic Cooperation and Development (1992) *OECD Statistical Information Research and Inquiry System Magnetic Tape Format Documentation*, Paris, France.

Organization For Economic Cooperation and Development (1992) *Quarterly National Accounts Inventory of Series Codes*, Paris, France.

Organization For Economic Cooperation and Development (1992) *Quarterly National Accounts Technical Documentation*, Paris, France.

Standard & Poor's Compustat Services Inc. (1991), *COMPUSTAT II Documentation*, Englewood, CO.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ETS User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1546 pp.

SAS/ETS User's Guide, Version 8

ISBN 1–58025–489–6

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999