THE POWER TO KNOW.

SAS/ETS® 13.2 User's Guide

The DATASOURCE Procedure

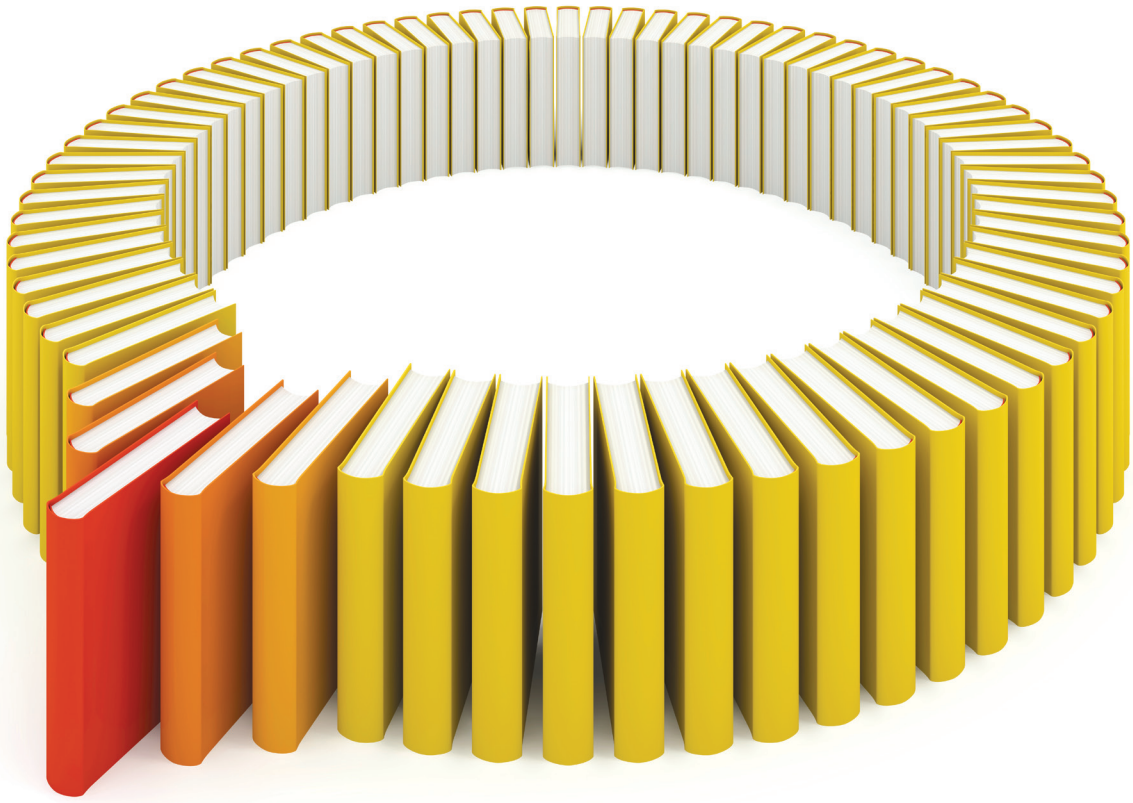
This document is an individual chapter from *SAS/ETS® 13.2 User's Guide*.

The correct bibliographic citation for the complete manual is as follows: SAS Institute Inc. 2014. *SAS/ETS® 13.2 User's Guide*. Cary, NC: SAS Institute Inc.

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

August 2014

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit **support.sas.com/bookstore** or call 1-800-727-3228.



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.

sas
THE POWER TO KNOW®

Chapter 12
The DATASOURCE Procedure

Contents

Overview: DATASOURCE Procedure

The DATASOURCE procedure extracts time series and event 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 and event 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 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= option names a data set containing information on time series variables, the OUTBY= option names a data set that reports information on cross-sectional variables, and the OUTALL= option names a data set that combines both time series variables and cross-sectional information.

In addition to the auxiliary data sets, two types of primary output data sets are the OUT= and OUTEVENT= data sets. The OUTEVENT= data set contains event variables but excludes periodic time series data. The OUT= data set contains periodic time series data and any event variables referenced in the KEEP statement.

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 the following:

- U.S. Bureau of Economic Analysis data files:
  - National Income and Product Accounts
  - National Income and Product Accounts PC format
  - S-pages
- 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
  - COMPUSTAT Full Coverage Annual
  - COMPUSTAT Full Coverage 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

- Global Insight, formerly DRI/McGraw-Hill data files:

  - Basic Economics Data (formerly CITIBASE)
  - DRI Data Delivery Service files
  - CITIBASE Data Files
  - DRI Data Delivery Service Time Series
  - PC Format CITIBASE Databases

- FAME Information Services Databases

- Haver Analytics data files

  - United States Economic Indicators
  - Specialized Databases
  - Financial Indicators
  - Industry
  - Industrial Countries
  - Emerging Markets
  - International Organizations
  - Forecasts and As Reported Data
  - United States Regional

- 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: DATASOURCE Procedure

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 in Table 12.1, extracted from a DRIBASIC data file, gives the foreign exchange rates for Japan, Switzerland, and the United Kingdom, respectively.

Table 12.1 The Form of SAS Data Sets Required by Most SAS/ETS Procedures

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

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 statement 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 contains only 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 by 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 and are shown in Figure 12.1.

```
/*-- Selecting Time Series Variables -- The KEEP and DROP Statements --*/
filename citifile "%sysget(DATASRC_DATA)citiaf.dat" RECFM=F LRECL=80;
proc datasource filetype=dribasic infile=citifile
                interval=month  outcont=vars;
   drop e: ;
run;

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

Figure 12.1 Listing of the OUTCONT= Data Set

Some Time Series Variables Available in CITIFILE

Obs	NAME	KEPT	SELECTED	TYPE	LENGTH	VARNUM	LABEL
1	BUS	1	1	1	5	.	INDEX OF NET BUSINESS FORMATION, (1967=100;SA)
2	CCBPY	1	1	1	5	.	RATIO, CONSUMER INSTAL CREDIT TO PERSONAL INCOME (%,SA)(BCD-95)
3	CCI30M	1	1	1	5	.	CONSUMER INSTAL.LOANS: DELINQUENCY RATE,30 DAYS & OVER, (%,SA)
4	CCIPY	1	1	1	5	.	RATIO, CONSUMER INSTAL CREDIT TO PERSONAL INCOME (%,SA)(BCD-95)
5	COCI77	1	1	1	5	.	CONSTRUCTION COST INDEX: DEPT OF COMMERCE COMPOSITE(1977=100,NSA)
6	CONU	1	1	1	5	.	CONSTRUCT.PUT IN PLACE: PRIV NEW HOUSING UNITS (MIL$,SAAR)
7	DLEAD	1	1	1	5	.	COMPOSITE INDEX OF 12 LEADING INDICATORS(67=100,SA)
8	F6CMB	1	1	1	5	.	DEPOSITORY INST RESERVES: TOTAL BORROWINGS AT RES BANKS(MIL$,NSA)
9	F6EDM	1	1	1	5	.	U.S.MDSE EXPORTS: MANUFACTURED GOODS (MIL$,NSA)
10	WTNO8	1	1	1	5	.	MFG & TRADE SALES:MERCHANT WHOLESALERS,OTHR NONDUR GDS,82$
11	WTNR	1	1	1	5	.	MERCHANT WHOLESALERS' SALES: NONDURABLE GOODS (MIL$,SA)
12	WTR	1	1	1	5	.	MERCHANT WHOLESALERS' SALES: TOTAL (MIL$,SA)

Obs	FORMAT	FORMATL	FORMATD	CODE
1		0	0	BUS
2		0	0	CCBPY
3		0	0	CCI30M
4		0	0	CCIPY
5		0	0	COCI77
6		0	0	CONU
7		0	0	DLEAD
8		0	0	F6CMB
9		0	0	F6EDM
10		0	0	WTNO8
11		0	0	WTNR
12		0	0	WTR

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. Figure 12.2 shows an example extracting the foreign exchange rates from September 1985 to February 1987, you can use the following statements:

```
/*-- Controlling the Time Range of Data - The RANGE Statement --*/
filename citifile "%sysget(DATASRC_DATA)citiaf.dat" RECFM=F LRECL=80;
proc datasource filetype=dribasic infile=citifile
                interval=month  out=dataset;
   keep  exrjan exrsw exruk;
   range from 1985:9 to 1987:2;
run;

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

Figure 12.2 Subset Obtained by KEEP and RANGE Statements

Printout of the OUT= Data Set

Obs	DATE	EXRJAN	EXRSW	EXRUK
1	SEP1985	236.530	2.37490	136.420
2	OCT1985	214.680	2.16920	142.150
3	NOV1985	204.070	2.13060	143.960
4	DEC1985	202.790	2.10420	144.470
5	JAN1986	199.890	2.06600	142.440
6	FEB1986	184.850	1.95470	142.970
7	MAR1986	178.690	1.91500	146.740
8	APR1986	175.090	1.90160	149.850
9	MAY1986	167.030	1.85380	152.110
10	JUN1986	167.540	1.84060	150.850
11	JUL1986	158.610	1.74450	150.710
12	AUG1986	154.180	1.66160	148.610
13	SEP1986	154.730	1.65370	146.980
14	OCT1986	156.470	1.64330	142.640
15	NOV1986	162.850	1.68580	142.380
16	DEC1986	162.050	1.66470	143.930
17	JAN1987	154.830	1.56160	150.540
18	FEB1987	153.410	1.54030	152.800

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 12.2 contains three cross sections:

- Cross-section one is identified by (COUNTRY='112' CSC='F' PARTNER=' ' VERSION='Z').
- Cross-section two is identified by (COUNTRY='146' CSC='F' PARTNER=' ' VERSION='Z').
- Cross-section three is identified by (COUNTRY='158' CSC='F' PARTNER=' ' VERSION='Z').

Table 12.2 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 Table 12.1 and Table 12.2 use two different ways of representing time series data for three different countries: the United Kingdom (COUNTRY='112'), Switzerland (COUNTRY='146'), and Japan (COUNTRY='158'). The first representation (Table 12.1) incorporates each country's name into the series names, while the second representation (Table 12.2) represents countries as different cross sections by using the BY variable named COUNTRY. See "Time Series and SAS Data Sets" in Chapter 3, "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, and are shown in Figure 12.3.

```
filename ifsfile "%sysget(DATASRC_DATA)imfifs1.dat" RECFM=F LRECL=88;
proc datasource
   filetype=imfifsp infile=ifsfile
   outselect=on ebcdic
   interval=month
   outby=xsection;
run;

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

Figure 12.3 Listing of the OUTBY= Data Set

Some Cross Sections Available in IFSFILE

Obs	COUNTRY	CSC	PARTNER	VERSION	ST_DATE	END_DATE	NTIME	NOBS	NSERIES	NSELECT	CNTYNAME
1	111	F		Z	JAN1957	SEP1986	357	357	6	3	UNITED STATES
2	112	F		Z	JAN1957	SEP1986	357	357	6	3	UNITED KINGDOM
3	146	F		Z	JAN1957	SEP1986	357	357	6	3	SWITZERLAND
4	158	F		Z	JAN1957	SEP1986	357	357	6	3	JAPAN
5	186	F		Z	JAN1957	SEP1986	357	357	6	3	TURKEY

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 all monthly series defined for all cross sections (COUNTRY='111', COUNTRY='112', COUNTRY='146', COUNTRY='158', and COUNTRY='186') which are shown in Figure 12.4.

```
filename datafile "%sysget(DATASRC_DATA)imfifs1.dat" RECFM=F LRECL=88;

title3 'Time Series Defined in Cross Section';
proc datasource filetype=imfifsp
   outselect=on ebcdic
   interval=month
   outall=ifsall;
run;

title4 'Cross Sections Available in OUTALL=IFSALL Data Set';
proc print
   data=ifsall;
run;
```

Figure 12.4 Listing of the OUTALL= Data Set

Some Cross Sections Available in IFSFILE

Time Series Defined in Cross Section
Cross Sections Available in OUTALL=IFSALL Data Set

Obs	COUNTRY	CSC	PARTNER	VERSION	NAME	KEPT	SELECTED	TYPE	LENGTH	VARNUM
1	111	F		Z	F___AA	1	1	1	5	.
2	111	F		Z	F___AC	1	1	1	5	.
3	111	F		Z	F___AE	1	1	1	5	.
4	112	F		Z	F___AA	1	1	1	5	.
5	112	F		Z	F___AC	1	1	1	5	.
6	112	F		Z	F___AE	1	1	1	5	.
7	146	F		Z	F___AA	1	1	1	5	.

Obs	BLKNUM	LABEL	FORMAT	FORMATL	FORMATD	ST_DATE	END_DATE	NTIME	NOBS
1	1	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
2	2	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
3	3	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
4	4	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
5	5	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
6	6	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
7	7	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357

Obs	CNTYNAME	SUBJECT	SCDATA	DATATYPE	DU_CODE	DU_NAME	NDEC	BASEYEAR	SOURCE
1	UNITED STATES		S	E	U	U	5		
2	UNITED STATES		S	F	U	U	5		
3	UNITED STATES		S	A	U	U	5		
4	UNITED KINGDOM		S	E	U	U	6		
5	UNITED KINGDOM		S	F	U	U	5		
6	UNITED KINGDOM		S	A	U	U	6		
7	SWITZERLAND		S	E	U		4		

Figure 12.4 *continued*

Some Cross Sections Available in IFSFILE

Time Series Defined in Cross Section
Cross Sections Available in OUTALL=IFSALL Data Set

Obs	COUNTRY	CSC	PARTNER	VERSION	NAME	KEPT	SELECTED	TYPE	LENGTH	VARNUM
8	146	F		Z	F___AC	1	1	1	5	.
9	146	F		Z	F___AE	1	1	1	5	.
10	158	F		Z	F___AA	1	1	1	5	.
11	158	F		Z	F___AC	1	1	1	5	.
12	158	F		Z	F___AE	1	1	1	5	.
13	186	F		Z	F___AA	1	1	1	5	.

Obs	BLKNUM	LABEL	FORMAT	FORMATL	FORMATD	ST_DATE	END_DATE	NTIME	NOBS
8	8	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
9	9	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
10	10	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
11	11	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
12	12	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
13	13	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357

Obs	CNTYNAME	SUBJECT	SCDATA	DATATYPE	DU_CODE	DU_NAME	NDEC	BASEYEAR	SOURCE
8	SWITZERLAND			S	F	U	6		
9	SWITZERLAND			S	A	U	4		
10	JAPAN			S	E	U	3		
11	JAPAN			S	F	U	6		
12	JAPAN			S	A	U	3		
13	TURKEY			S	E	U	3		

Figure 12.4 *continued*

Some Cross Sections Available in IFSFILE

Time Series Defined in Cross Section
Cross Sections Available in OUTALL=IFSALL Data Set

Obs	COUNTRY	CSC	PARTNER	VERSION	NAME	KEPT	SELECTED	TYPE	LENGTH	VARNUM
14	186	F		Z	F___AC	1	1	1	5	.
15	186	F		Z	F___AE	1	1	1	5	.

Obs	BLKNUM	LABEL	FORMAT	FORMATL	FORMATD	ST_DATE	END_DATE	NTIME	NOBS
14	14	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357
15	15	MARKET RATE CONVERSION FACTOR		0	0	JAN1957	SEP1986	357	357

Obs	CNTYNAME	SUBJECT	SCDATA	DATATYPE	DU_CODE	DU_NAME	NDEC	BASEYEAR	SOURCE
14	TURKEY		S	F	U		5		
15	TURKEY		S	A	U		3		

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 of available data. 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 condition that the BY variables must satisfy for a cross section to be selected. For example, the following statements extract the monthly market rate conversion factors for the United Kingdom (COUNTRY='112') and Switzerland (COUNTRY='146') for the period from September 1985 to February 1986.

```
filename datafile "%sysget(DATASRC_DATA)imfifs1.dat" RECFM=F LRECL=88;

title3 'Time Series Defined in Selected Cross Sections';
proc datasource filetype=imfifsp
   outselect=on ebcdic
   interval=month
   out=ifs;

   where country in ('146', '112') and partner=' ';
   keep F___AA F___AC;
   range from '01sep85'd to '01feb86'd;
run;
```

```
title4 'OUTALL=IFS Data Set';
proc print
   data=ifs;
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 market rate conversion factor (F___AA) and market rate conversion factor (F___AC) used by IMF can be given more descriptive names and labels by the following statements and are shown in Figure 12.5 and Figure 12.6.

```
filename ifsfile "%sysget(DATASRC_DATA)imfifs1.dat" RECFM=F LRECL=88;

proc datasource filetype=imfifsp infile=ifsfile
                interval=month
                out=market outcont=mrktvars;
   where country in ('112','146','158') and partner=' ';
   keep  f___aa f___ac;
   range from '01jun85'd to '01feb86'd;
   rename  f___aa=alphmkt  f___ac=charmkt;
   label   f___aa='F___AA: Market Rate Conversion Factor Used in Alpha Test'
           f___ac='F___AC: Market Rate Conversion Used in Charlie Test';
run;

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

title1 'Contents of OUT= Showing New NAMEs and LABELs';
proc contents data=market;
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 in 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" on page 663 section.

Figure 12.5 Renaming and Labeling Variables

Printout of OUTCONT= Showing New NAMEs and LABELs

Obs	NAME	LABEL	LENGTH
1	alphmkt	F___AA: Market Rate Conversion Factor Used in Alpha Test	5
2	charmkt	F___AC: Market Rate Conversion Used in Charlie Test	5

Figure 12.6 Renaming and Labeling Variables

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	COUNTRY	Char	3		COUNTRY CODE
2	CSC	Char	1		CONTROL SOURCE CODE
5	DATE	Num	4	MONYY7.	Date of Observation
3	PARTNER	Char	3		PARTNER COUNTRY CODE
4	VERSION	Char	1		VERSION CODE
6	alphmkt	Num	5		F___AA: Market Rate Conversion Factor Used in Alpha Test
7	charmkt	Num	5		F___AC: Market Rate Conversion Used in Charlie Test

Notice that even though you changed the names of F___AA and F___AC to alphmkt and charmkt, respectively, you still use 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 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. Output is shown in Figure 12.7 and Figure 12.8.

```
filename ifsfile "%sysget(DATASRC_DATA)imfifs1.dat" RECFM=F LRECL=88;

proc datasource filetype=imfifsp infile=ifsfile
                interval=month
                out=market outcont=mrktvars;
   where country in ('112','146','158') and partner=' ';
   keep  f___aa f___ac;
   range from '01jun85'd to '01feb86'd;
   rename  f___aa=alphmkt  f___ac=charmkt;
   label   f___aa='F___AA: Market Rate Conversion Factor Used in Alpha Test'
           f___ac='F___AC: Market Rate Conversion Used in Charlie Test';
   length _numeric_ 4;
   length f___aa 6;
run;

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

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

Figure 12.7 Changing the Lengths of Numeric Variables

Printout of OUTCONT= Showing New NAMEs and LABELs

Obs	NAME	LABEL	LENGTH
1	alphmkt	F___AA: Market Rate Conversion Factor Used in Alpha Test	6
2	charmkt	F___AC: Market Rate Conversion Used in Charlie Test	4

Figure 12.8 Changing the Lengths of Numeric Variables

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	COUNTRY	Char	3		COUNTRY CODE
2	CSC	Char	1		CONTROL SOURCE CODE
5	DATE	Num	4	MONYY7.	Date of Observation
3	PARTNER	Char	3		PARTNER COUNTRY CODE
4	VERSION	Char	1		VERSION CODE
6	alphmkt	Num	6		F___AA: Market Rate Conversion Factor Used in Alpha Test
7	charmkt	Num	4		F___AC: Market Rate Conversion Used in Charlie Test

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

Syntax: DATASOURCE Procedure

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, subsetting statements and attribute statements. Subsetting statements provide selection of time series data over selected time periods and cross sections from the input data file. Attribute statements 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 12.3.

Table 12.3 Summary of Syntax

Option	Description
Input Data File Options	
FILETYPE=	type of input data file to read
INFILE=	fileref(s) of the input data
LRECL=	lrecl(s) of the input data
RECFM=	recfm(s) of the input data
ASCII	character set of the incoming data
EBCDIC	character set of the incoming data
Output Data Set Options	
OUT=	write the extracted time series data
OUTALL=	information on time series and cross sections
OUTBY=	information on only cross sections
OUTCONT=	information on only time series variables
OUTEVENT=	write event-oriented data
OUTSELECT=	control reporting of all or only selected series and cross sections
INDEX	create single indexes from BY variables for the OUT= data set
ALIGN=	control the alignment of SAS date values
Subsetting Option and Statements	
INTERVAL=	select periodicity of series to extract
KEEP	time series to include in the OUT= data set
DROP	time series to exclude from the OUT= data set
KEEPEVENT	events to include in the OUTEVENT= data set
DROPEVENT	events to exclude from the OUTEVENT= data set
WHERE	select cross sections for output
RANGE	time range of observations to be output
Assigning Attributes Options and Statements	
FORMAT	assign formats to variables in the output data sets
ATTRIBUTE FORMAT=	assign formats to variables in the output data sets
LABEL	assign labels to variables in the output data sets
ATTRIBUTE LABEL=	assign labels to variables in the output data sets
LENGTH	control the lengths of variables in the output data sets
ATTRIBUTE LENGTH=	control the lengths of variables in the output data sets
RENAME	assign new names to variables in the output data sets

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 used 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 in 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 "Data Elements Reference: DATASOURCE Procedure" on page 690 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: Concepts* 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 the 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 Statements: Reference* for more details on the FILENAME statement. Physical data files can reside on DVD, 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 "Data Elements Reference: DATASOURCE Procedure" on page 690.

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 "Data Elements Reference: DATASOURCE Procedure" on page 690.

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 "Data Elements Reference: DATASOURCE Procedure" on page 690. 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 can 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 "Data Elements Reference: DATASOURCE Procedure" on page 690 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 "OUT= Data Set" on page 685 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 "OUTALL= Data Set" on page 689 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 are written to the OUTBY= data set. If you omit the OUTBY= option, no OUTBY= data set is created. See "OUTBY= Data Set" on page 688 for further details.

OUTCONT= *SAS-data-set*
writes information on the contents of the input data file to an output data set. By default, 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 "OUTCONT= Data Set" on page 687 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 "OUTEVENT= Data Set" on page 689 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 is 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 "Variable Lists" on page 684 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 respects:

- 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 variables different from 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 “Variable Lists” on page 684 for details.

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

There is a default DROP or 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 variables different from 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. 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 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 "Variable Lists" on page 684 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 respects:

- 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. 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 "Variable Lists" on page 684 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 respects:

- 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 "Data Elements Reference: DATASOURCE Procedure" on page 690 lists the names of the BY variables for each file type.

For example, DOTS (Direction of Trade Statistics) files, distributed by the 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: Concepts* 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 two-digit year is specified, PROC DATASOURCE uses 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 two-digit year specification.

The default YEARCUTOFF value is 1926. To use a different YEARCUTOFF value, specify

```
options yearcutoff=yyyy;
```

where YYYY is the YEARCUTOFF value you want to use. See *SAS System Options: Reference* 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 are 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 are available, you can first run PROC DATASOURCE with the OUTBY= or OUTALL= option. The 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 "Variable Lists" on page 684 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.

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 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 "Variable Lists" on page 684 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.

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 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 32 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" as "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: DATASOURCE Procedure

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, such as 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_. The keyword _NUMERIC_ specifies all numeric variables, _CHARACTER_ specifies all character variables, and _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 either to 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 "Data Elements Reference: DATASOURCE Procedure" on page 690 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 PW112 and PW117 themselves, and a single variable PZU. The second statement drops all the variables that fall alphabetically between DATA1 and DATA99, and between 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. Variable lists can not exceed 200 characters in length.

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: Concepts* 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 Table 12.4.

Table 12.4 The OUT= Data Set Containing Unique Series for Each BY Group

<table>
<tr><th>BY
Variables
BY1 . . . BYP</th><th>Series in
first BY group
F1 F2 F3 . . . FN</th><th>Series in
second BY group
S1 S2 S3 . . . SM</th><th>. . .
. . .
. . .</th><th>Series in
last BY group
T1 T2 T3 . . . TK</th></tr>
<tr><td>BY
group
1</td><td>DATA
is
here</td><td colspan="3" rowspan="3"></td></tr>
<tr><td>BY
group
2</td><td rowspan="4"></td><td>DATA
is
here</td><td colspan="2">data is missing
everywhere except
on diagonal</td></tr>
<tr><td>⋮</td><td></td><td>DATA
is
here</td><td rowspan="2"></td></tr>
<tr><td>BY
group
N</td><td colspan="2"></td><td>DATA
is
here</td></tr>
</table>

The data in Table 12.4 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. KEPT is usually the same as SELECTED, but can 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 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 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 the 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 can vary from one series to another and 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 can 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, which are included in the OUTEVENT= data set only if they have data in at least one selected BY group, and 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 cannot contain any periodic time series variables, the OUT= data set can contain event variables if they are explicitly specified in a KEEP statement. In summary, 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, so in summary, the WHERE and the RANGE statements are effective for both output data sets.

Data Elements Reference: DATASOURCE Procedure

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 composing 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 “References”.) The currently supported file types are summarized in Table 12.5.

Table 12.5 Supported File Types

Supplier	FILETYPE=	Description
BEA	BEANIPA	National Income and Product Accounts
	BEANIPAD	National Income and Product Accounts PC 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

Table 12.5 *continued*

Supplier	FILETYPE=	Description
GLOBAL INSIGHT	DRIBASIC	Basic Economic (formerly CITIBASE) Data Files
	CITIBASE	CITIBASE Data Files
(DRI)	DRIDDS	DRI Data Delivery Service Time Series
(DRI)	CITIDISK	PC Format CITIBASE Databases
CRSP	CRY2DBS	Y2K Daily Binary Security File Format
	CRY2DBI	Y2K Daily Binary Calendar&Indices File Format
	CRY2DBA	Y2K Daily Binary File Annual Data Format
	CRY2MBS	Y2K Monthly Binary Security File Format
	CRY2MBI	Y2K Monthly Binary Calendar&Indices File Format
	CRY2MBA	Y2K Monthly Binary File Annual Data Format
	CRY2DCS	Y2K Daily Character Security File Format
	CRY2DCI	Y2K Daily Character Calendar&Indices File Format
	CRY2DCA	Y2K Daily Character File Annual Data Format
	CRY2MCS	Y2K Monthly Character Security File Format
	CRY2MCI	Y2K Monthly Character Calendar&Indices File Format
	CRY2MCA	Y2K Monthly Character File Annual Data Format
	CRY2DIS	Y2K Daily IBM Binary Security File Format
	CRY2DII	Y2K Daily IBM Binary Calendar&Indices File Format
	CRY2DIA	Y2K Daily IBM Binary File Annual Data Format
	CRY2MIS	Y2K Monthly IBM Binary Security File Format
	CRY2MII	Y2K Monthly IBM Binary Calendar&Indices File Format
	CRY2MIA	Y2K Monthly IBM Binary File Annual Data Format
	CRY2MVS	Y2K Monthly VAX Binary Security File Format
	CRY2MVI	Y2K Monthly VAX Binary Calendar&Indices File Format
	CRY2MVA	Y2K Monthly VAX Binary File Annual Data Format
	CRY2DVS	Y2K Daily VAX Binary Security File Format
	CRY2DVI	Y2K Daily VAX Binary Calendar&Indices File Format
	CRY2DVA	Y2K Daily VAX Binary File Annual Data Format
	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
	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
	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

Table 12.5 *continued*

Supplier	FILETYPE=	Description
CRSP	CRSPMII	CRSP Monthly IBM Binary Calendar&Indices File Format
	CRSPMIA	CRSP Monthly IBM Binary File Annual Data Format
	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
	CRSPDVI	CRSP Daily VAX Binary Calendar&Indices File Format
	CRSPDVA	CRSP Daily VAX Binary File Annual Data Format
	CRSPMUS	CRSP Monthly UNIX Binary Security File Format or utility dump of CRSPAccess Monthly Security File Format
	CRSPMUI	CRSP Monthly UNIX Binary Calendar&Indices File Format or utility dump of CRSPAccess Monthly Cal&Indices Format
	CRSPMUA	CRSP Monthly UNIX Binary File Annual Data Format or utility dump of CRSPAccess Monthly Annual Data Format
	CRSPDUS	CRSP Daily UNIX Binary Security File Format or utility dump of CRSPAccess Daily Security Format
	CRSPDUI	CRSP Daily UNIX Binary Calendar&Indices File Format or utility dump of CRSPAccess Daily Calendar&Indices Format
	CRSPDUA	CRSP Daily UNIX Binary File Annual Data Format or utility dump of CRSPAccess Daily Annual Data 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
	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
	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
	CR95MUS	CRSP 1995 Monthly UNIX Binary Security File Format
	CR95MUI	CRSP 1995 Monthly UNIX Binary Calendar&Indices File Format
	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

Table 12.5 *continued*

Supplier	FILETYPE=	Description
	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
	CR95MAS	CRSP 1995 Monthly ALPHA Binary Security File Format
	CR95MAI	CRSP 1995 Monthly ALPHA Binary Calendar&Indices 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
FAME	FAME	FAME Information Services Databases
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 Format
	OECDQNA	OECD Quarterly National Accounts Format
	OECDMEI	OECD Main Economic Indicators 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
	CSAIY2	Y2K COMPUSTAT Annual, IBM 360&370 Format
	CSQIY2	Y2K COMPUSTAT 48 Quarter, IBM 360&370 Format
	CSAUCY2	Y2K COMPUSTAT Annual, Universal Character Format
	CSQUCY2	Y2K COMPUSTAT 48 Quarter, Universal Character Format

Data supplier abbreviations used in Table 12.5 are summarized in Table 12.6.

Table 12.6 Data Supplier Abbreviations

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	Global Insight (formerly DRI/McGraw-Hill)
FAME	FAME Information Services, Inc.
GLOBAL INSIGHT	Global Insight, 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 macroeconomic 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 Format

Table 12.7 FILETYPE=BEANIPA–National Income and Product Accounts Format

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	

Table 12.7 (BEANIPA–National Income and Product Accounts Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
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 PC Format

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

Table 12.8 FILETYPE=BEANIPAD–National Income and Product Accounts PC Format

Metadata Field Types	Metadata Fields	Metadata Labels
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.	

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)

Table 12.9 FILETYPE=BLSCPI–Consumer Price Index Surveys (=CU,CW)

Metadata Field Types	Metadata Fields	Metadata Labels
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.	
Missing Codes	A data value of 0 is interpreted as MISSING.	

FILETYPE=BLSWPI–Producer Price Index Survey (WP)

Table 12.10 FILETYPE=BLSWPI–Producer Price Index Survey (WP)

Metadata Field Types	Metadata Fields	Metadata Labels
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

Table 12.11 FILETYPE=BLSEENA–National Employment, Hours, and Earnings Survey

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	Database is stored in a single file.	
INTERVAL=	YEAR, QUARTER, MONTH (default)	

Table 12.11 (BLSEENA–National Employment, Hours, and Earnings Survey Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
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

Table 12.12 FILETYPE=BLSEESA–State and Area Employment, Hours, and Earnings Survey

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	Database is stored in a single 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.	

Global Insight DRI 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.

Global Insight, formerly DRI/McGraw-Hill, distributes Basic Economic data files on various media. Old DRIDDS data files can be read by DATASOURCE using the DRIDDS filetype.

The following DRI file types are supported.

FILETYPE=DRIBASIC–Global Insight DRI Basic Economic Data Files

Table 12.13 FILETYPE=DRIBASIC–Global Insight DRI Basic Economic Data Files

Metadata Field Types	Metadata Fields	Metadata Labels
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–Global Insight DRI Data Delivery Service Data Files

Table 12.14 FILETYPE=DRIDDS–Global Insight DRI Data Delivery Service Data Files

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), SEMIYEAR, QUARTER, MONTH, SEMIMONTH, 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 distributed prior to May 1987.

Table 12.15 FILETYPE=CITIOLD–Old Format CITIBASE Data Files

Metadata Field Types	Metadata Fields	Metadata Labels
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 Format CITIBASE Databases

Table 12.16 FILETYPE=CITIDISK–PC Format CITIBASE Databases

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	Database is stored in groups of three associated files having the same filename 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	

Table 12.16 (CITIDISK–PC Format CITIBASE Databases Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
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 & 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 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
FILETYPE=CSAIY2–Four-Digit Year COMPUSTAT Annual, IBM 360/370 Format

Table 12.17 FILETYPE=CSAIBM,CSAIY2 –COMPUSTAT Annual,IBM 360/370 Format

Metadata Field Types	Metadata Fields	Metadata Labels
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)

Table 12.17 CSAIBM,CSAIY2 –COMPUSTAT Annual,IBM 360/370 Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
	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 DATA322-DATA326 DATA338 DATA345-DATA347 DATA350 AFTNT52-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
FILETYPE=CSQIY2–FOUR-DIGIT YEAR COMPUSTAT 48-Quarter, IBM 360/370 Format

Table 12.18 FILETYPE=CS48QIBM,CSQIY2 –COMPUSTAT 48-Quarter, IBM 360/370 Format

Metadata Field Types	Metadata Fields	Metadata Labels
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)

Table 12.18 CS48QIBM,CSQIY2 –COMPUSTAT 48-Quarter, IBM 360/370 Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
	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
FILETYPE=CSAUCY2–Four-Digit Year COMPUSTAT Annual, Universal Character Format

Table 12.19 FILETYPE=CSAUC,CSAUCY2 –COMPUSTAT Annual, Universal Character Format

Metadata Field Types	Metadata Fields	Metadata Labels
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 DATA322-DATA326 DATA338 DATA345-DATA347 DATA350 AFTNT52-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
FILETYPE=CSQUCY2–Four-Digit Year COMPUSTAT 48 Quarter, Universal Character Format

Table 12.20 FILETYPE=CS48QUC,CSQUCY2 –COMPUSTAT 48 Quarter, Universal Character Format

Metadata Field Types	Metadata Fields	Metadata Labels
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

Table 12.20 CS48QUC,CSQUCY2 –COMPUSTAT 48 Quarter, Universal Character Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
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 through two primary stock files, the NYSE/AMEX file and the NASDAQ file. These files contain 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 are distributed in CRSPAccess format. See Chapter 38, "The SASECRSP Interface Engine," for more about accessing your CRSPAccess database. You can convert your CRSPAccess data to binary format (SFA format) by using the CRSP-supplied utility (STK_DUMP_BIN). Use the DATASOURCE procedure for SFA format access and use SASECRSP Interface for CRSPAccess.

CRSP stock data (in SFA format) 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.

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 data 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 used 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 cannot do the conversion on one host and transfer and read the file on another host.

If you are using the CRSPAccess 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.

For the four-digit year data, use the Y2K-compliant filetypes for that data type.

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 *security1* < *security2* . . . > are the filerefs given to the security files, in the order in which they should be read.

CRSP Calendar/Indices Files

Table 12.21 CRSP Calendar/Indices Files Format

Metadata Field Types	Metadata Fields	Metadata Labels
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

Table 12.22 CRSP Daily Security Files Format

Metadata Field Types	Metadata Fields	Metadata Labels
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)

Table 12.22 CRSP Daily Security Files Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels	
	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

Table 12.23 CRSP Monthly Security Files Format

Metadata Field Types	Metadata Fields	Metadata Labels	
Data Files	INFILE=(calfile security1 < security2 . . . >)		
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

Table 12.23 CRSP Monthly Security Files Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels	
		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 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 Annual Data

Table 12.24 CRSP Annual Data Format

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	INFILE=(security1 < security2 . . . >)	
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 KEEP Lists	All variables will be kept.	

FAME Information Services Databases

The DATASOURCE procedure provides access to FAME Information Services databases for UNIX-based systems only. See "The SASEFAME Interface Engine" in Chapter 41, "The SASEFAME Interface Engine," for information about a more flexible FAME database access.

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 by giving the following specifications.

Specify FILETYPE=FAME in the PROC DATASOURCE statement and give the FAME database name to access with a DBNAME='*fame-database* ' option. The character string you specify in 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.

Name the FAME series to be extracted with a KEEP statement. The items in the KEEP statement are passed through to FAME software; therefore, you can use any valid FAME expression to specify the series to be extracted. Enclose in quotes any FAME series name or expression that is not a valid SAS name.

Name the SAS variable names you want to use for the extracted series in 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 might 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 in 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 SASEFAME Interface Engine" in Chapter 41, "The SASEFAME Interface Engine."

FILETYPE=FAME–FAME Information Services Databases

Table 12.25 FILETYPE=FAME–FAME Information Services Database Format

Metadata Field Types	Metadata Fields	Metadata Labels
INTERVAL=	YEAR	correspond 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)

Table 12.25 FILETYPE=FAME–FAME Information Services Database Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
	YEAR.11	correspond to FAME's ANNUAL(OCTOBER)
	YEAR.12	correspond to FAME's ANNUAL(NOVEMBER)
	SEMIYEAR	correspond to FAME's SEMIYEAR
	QUARTER	correspond to FAME's QUARTER
	MONTH	correspond to FAME's MONTH
	SEMIMONTH	correspond to FAME's SEMIMONTH
	TENDAY	correspond to FAME's TENDAY
	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	correspond to FAME's WEEKDAY
	DAY	correspond to FAME's DAY
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 United States and other countries. See "The SASEHAVR Interface Engine" in Chapter 42, "The SASEHAVR Interface Engine," for information about accessing your HAVER DLX database. SASEHAVR is supported on most Windows environments. Use the DATASOURCE procedure for serial access of your data. The format of Haver Analytics data files is similar to the CITIBASE/DRIBASIC formats.

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

Table 12.26 FILETYPE=HAVER–Haver Analytics Data Files Format

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	Database is stored in a single file.	
INTERVAL=	YEAR (default), QUARTER, MONTH	
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 subscriptions for their International Financial Statistics (IFS), Direction of Trade Statistics (DOTS), Balance of Payment Statistics (BOPS), and Government Finance Statistics (GFS) databases. The first three contain annual, quarterly, and monthly data, while the GFS file has only annual data.

PROC DATASOURCE supports only the packed format IMF data.

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.

Table 12.27 FILETYPE=IMFIFSP–International Financial Statistics Format

Metadata Field Types	Metadata Fields	Metadata Labels
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.

Table 12.28 FILETYPE=IMFDOTSP–Direction of Trade Statistics Format

Metadata Field Types	Metadata Fields	Metadata Labels
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.

Table 12.29 FILETYPE=IMFBOPSP–Balance of Payment Statistics Format

Metadata Field Types	Metadata Fields	Metadata Labels
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.

Table 12.30 FILETYPE=IMFGFSP–Government Finance Statistics Format

Metadata Field Types	Metadata Fields	Metadata Labels
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).

Table 12.31 FILETYPE=OECDANA–Annual National Accounts Format

Metadata Field Types	Metadata Fields	Metadata Labels
Data Files	Database is stored on a single 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.	
Series Renamed	OLDNAME	NEWNAME
	p0discgdpe	p0digdpe
	doll2gdpe	dol2gdpe
	doll3gdpe	dol3gdpe
	doll1gdpe	dol1gdpe
	ppp1gdpd	pp1gdpd
	ppp1gdpd1	pp1gdpd1
	p0itxgdpc	p0itgdpc

Table 12.31 FILETYPE=OECDANA–Annual National Accounts Format continued)

Metadata Field Types	Metadata Fields	Metadata Labels
	p0itxgdps	p0itgdps
	p0subgdpc	p0sugdpc
	p0subgdps	p0sugdps
	p0cfcgdpc	p0cfgdpc
	p0cfgddps	p0cfgdps
	p0discgdpc	p0dicgdc
	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.

Table 12.32 FILETYPE=OECDQNA–Quarterly National Accounts Format

Metadata Field Types	Metadata Fields	Metadata Labels
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*.

Table 12.33 FILETYPE=OECDMEI–Main Economic Indicators Format

Metadata Field Types	Metadata Fields	Metadata Labels
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: DATASOURCE Procedure

Example 12.1: 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 (NIPA) file.

From the "Statistical Tables" published by the United States Department of Commerce, Bureau of Economic Analysis, the relation of foreign transactions in the Balance of Payments Accounts (BPA) are given in the fifth table (TABNUM='05') of the "Foreign Transactions" section (PARTNO='4'). Moreover, the first line in the table gives BPAs, while the eighth gives exports of goods and services. The series names __00100 and __00800, are constructed by two underscores followed by three digits as the line numbers, and then two digits as the column numbers.

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

```
/*- assign fileref to the external file to be processed --------*/

filename ascifile "%sysget(DATASRC_DATA)beanipa.data" recfm=v lrecl=108;

title1 'Relation of Foreign Transactions to Balance of Payment Accounts';
title2 'Range from 1984 to 1989';
```

```
title3 'Annual';
proc datasource filetype=beanipa infile=ascifile
                interval=year
                outselect=off
                outkey=byfor4;

   range from 1984 to 1989;
   keep __00100 __00800;

   label __00100='Balance of Payment Accounts';
   label __00800='Exports of Goods and Services';

   rename __00100=BPAs __00800=exports;
run;

proc print data=byfor4;
run;

 /*- assign fileref to the external file to be processed --------*/

filename ascifile "%sysget(DATASRC_DATA)beanipa.data" recfm=v lrecl=108;

title1 'Relation of Foreign Transactions to Balance of Payment Accounts';
title2 'Range from 1984 to 1989';

title3 'Annual';
proc datasource filetype=beanipa infile=ascifile
                interval=year
                outselect=off
                outkey=byfor4
                out=foreign4;

   range from 1984 to 1989;
   keep __00100 __00800;

   label __00100='Balance of Payment Accounts';
   label __00800='Exports of Goods and Services';

   rename __00100=BPAs __00800=exports;

run;

proc contents data=foreign4;
run;
proc print data=foreign4;
run;
```

The results are shown in Output 12.1.1, Output 12.1.2, and Output 12.1.3.

Output 12.1.1 Listing of OUTBY=byfor4 of the BEANIPA Data

Relation of Foreign Transactions to Balance of Payment Accounts
Range from 1984 to 1989
Annual

Obs	PARTNO	TABNUM	ST_DATE	END_DATE	NTIME	NOBS	NINRANGE	NSERIES	NSELECT
1	1	07	1929	1989	61	0	6	2	0
2	1	14	1929	1989	61	0	6	1	0
3	1	15	1929	1989	61	0	6	1	0
4	1	20	1967	1989	23	23	6	2	1
5	1	23	1929	1989	61	0	6	2	0
6	2	04	1929	1989	61	0	6	1	0
7	2	05	1929	1989	61	0	6	2	0
8	3	05	1929	1989	61	0	6	1	0
9	3	14	1952	1989	38	0	6	2	0
10	3	15	1952	1989	38	0	6	7	0
11	3	16	1952	1989	38	0	6	1	0
12	4	05	1946	1989	44	44	6	1	1
13	5	07	1929	1989	61	0	6	1	0
14	5	09	1929	1989	61	0	6	1	0
15	6	04	1929	1989	61	0	6	3	0
16	6	05	1929	1948	20	0	0	2	0
17	6	07	1929	1948	20	0	0	1	0
18	6	08	1929	1989	61	0	6	3	0
19	6	09	1948	1989	42	0	6	1	0
20	6	10	1929	1948	20	0	0	1	0
21	6	14	1929	1948	20	0	0	1	0
22	6	19	1929	1948	20	0	0	1	0
23	6	20	1929	1989	61	0	6	2	0
24	6	22	1929	1989	61	0	6	2	0
25	6	23	1948	1989	42	0	6	1	0
26	6	24	1948	1989	42	0	6	1	0
27	7	09	1929	1989	61	0	6	1	0
28	7	10	1929	1989	61	0	6	2	0
29	7	13	1959	1989	31	0	6	1	0

Output 12.1.2 CONTENTS of OUT=foreign4 of the BEANIPA Data

Relation of Foreign Transactions to Balance of Payment Accounts
Range from 1984 to 1989
Annual

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
3	DATE	Num	4	YEAR4.	Date of Observation
1	PARTNO	Char	1		Part Number of Publication, IntegerPortion of the Table Number, 1-9
2	TABNUM	Char	2		Table Number Within Part, DecimalPortion of the Table Number, 1-24
4	exports	Num	5		Exports of Goods and Services

Output 12.1.3 Listing of OUT=foreign4 of the BEANIPA Data

Relation of Foreign Transactions to Balance of Payment Accounts
Range from 1984 to 1989
Annual

Obs	PARTNO	TABNUM	DATE	exports
1	1	20	1984	44
2	1	20	1985	53
3	1	20	1986	46
4	1	20	1987	40
5	1	20	1988	48
6	1	20	1989	47
7	4	05	1984	3835
8	4	05	1985	3709
9	4	05	1986	3965
10	4	05	1987	4496
11	4	05	1988	5520
12	4	05	1989	6262

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 12.2: 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 the 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).

```
options yearcutoff = 1900;

filename datafile "%sysget(DATASRC_DATA)blscpi1.data" recfm=v lrecl=152;
proc datasource filetype=blscpi
   interval=mon
   outselect=off
   outby=cpikey(where=( upcase(areaname)
                  in ('NORTHEAST','NORTH CENTRAL','SOUTH','WEST')) )
   outcont=cpicont(where= ( index( upcase(label), 'MEDICAL CARE' )) );
   where survey='CU';
```

```
run;

title1 'OUTBY= Data Set, By AREANAME Selection';
proc print
   data=cpikey;
run;

title1 'OUTCONT= Data Set, By LABEL Selection';
proc print
   data=cpicont;
run;
```

The OUTBY= data set in Output 12.2.1 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 in Output 12.2.2 gives the variable names for medical care related series.

Output 12.2.1 Partial Listings of the OUTBY= Data Set

OUTBY= Data Set, By AREANAME Selection

Obs	SURVEY	SEASON	AREA	BASPTYPE	BASEPER	BYSELECT	ST_DATE
1	CU	U	0200	S	1982-84=100	1	DEC1977
2	CU	U	0100	S	1982-84=100	1	.
3	CW	U	0400	S	1982-84=100	0	DEC1977
4	CW	U	0100	S	1982-84=100	0	.
5	CW	U	0200	S	1982-84=100	0	.

Obs	END_DATE	NTIME	NOBS	NSERIES	NSELECT	SURTITLE	AREANAME
1	JUL1990	152	152	2	2	ALL URBAN CONSUM	NORTH CENTRAL
2	.	.	0	0	0	ALL URBAN CONSUM	NORTHEAST
3	JUL1990	152	0	1	0	URBAN WAGE EARN	WEST
4	.	.	0	0	0	URBAN WAGE EARN	NORTHEAST
5	.	.	0	0	0	URBAN WAGE EARN	NORTH CENTRAL

Output 12.2.2 Partial Listings of the OUTCONT= Data Set

OUTCONT= Data Set, By LABEL Selection

Obs	NAME	SELECTED	TYPE	LENGTH	VARNUM	LABEL	FORMAT	FORMATL	FORMATD
1	ASL5	1	1	5	.	SERVICES LESS MEDICAL CARE		0	0
2	A512	1	1	5	.	MEDICAL CARE SERVICES		0	0
3	A0L5	0	1	5	.	ALL ITEMS LESS MEDICAL CARE		0	0

The following statements make use of this information to extract the data for A512 and descriptive information on cross sections containing A512. Output 12.2.3 and Output 12.2.4 show these results.

```
options yearcutoff = 1900;

filename datafile "%sysget(DATASRC_DATA)blscpi1.data" recfm=v lrecl=152;

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

proc datasource filetype=blscpi interval=month
             out=medical outall=medinfo;
   where survey='CU' and area in ( '0100','0200','0300','0400' );
   keep date a512;
   range  from 1988:9;
   format area $areafmt.;
   rename a512=medcare;
run;

title1 'Information on Medical Care Service, OUTALL= Data Set';
proc print
   data=medinfo;
run;

title1 'Medical Care Service By Region, OUT= Data Set';
title2 'Range from September, 1988';
proc print
   data=medical;
run;
```

Output 12.2.3 Printout of the OUTALL= Data Set

Information on Medical Care Service, OUTALL= Data Set

Obs	SURVEY	SEASON	AREA	BASPTYPE	BASEPER	BYSELECT	NAME	KEPT	SELECTED	TYPE
1	CU	U	North Central Region	S	1982-84=100	1	medcare	1	1	1

Obs	LENGTH	VARNUM	BLKNUM	LABEL	FORMAT	FORMATL	FORMATD	ST_DATE	END_DATE	NTIME
1	5	7	50	MEDICAL CARE SERVICES		0	0	DEC1977	JUL1990	152

Obs	NOBS	NINRANGE	SURTITLE	AREANAME	S_CODE	UNITS	NDEC
1	152	23	ALL URBAN CONSUM	NORTH CENTRAL	CUUR0200SA512		1

Output 12.2.4 Printout of the OUT= Data Set

Medical Care Service By Region, OUT= Data Set
Range from September, 1988

Obs	SURVEY	SEASON	AREA	BASPTYPE	BASEPER	DATE	medcare
1	CU	U	North Central Region	S	1982-84=100	SEP1988	1364
2	CU	U	North Central Region	S	1982-84=100	OCT1988	1365
3	CU	U	North Central Region	S	1982-84=100	NOV1988	1368
4	CU	U	North Central Region	S	1982-84=100	DEC1988	1372
5	CU	U	North Central Region	S	1982-84=100	JAN1989	1387
6	CU	U	North Central Region	S	1982-84=100	FEB1989	1399
7	CU	U	North Central Region	S	1982-84=100	MAR1989	1405
8	CU	U	North Central Region	S	1982-84=100	APR1989	1413
9	CU	U	North Central Region	S	1982-84=100	MAY1989	1416
10	CU	U	North Central Region	S	1982-84=100	JUN1989	1425
11	CU	U	North Central Region	S	1982-84=100	JUL1989	1439
12	CU	U	North Central Region	S	1982-84=100	AUG1989	1452
13	CU	U	North Central Region	S	1982-84=100	SEP1989	1460
14	CU	U	North Central Region	S	1982-84=100	OCT1989	1473
15	CU	U	North Central Region	S	1982-84=100	NOV1989	1481
16	CU	U	North Central Region	S	1982-84=100	DEC1989	1485
17	CU	U	North Central Region	S	1982-84=100	JAN1990	1500
18	CU	U	North Central Region	S	1982-84=100	FEB1990	1516
19	CU	U	North Central Region	S	1982-84=100	MAR1990	1528
20	CU	U	North Central Region	S	1982-84=100	APR1990	1538
21	CU	U	North Central Region	S	1982-84=100	MAY1990	1548
22	CU	U	North Central Region	S	1982-84=100	JUN1990	1557
23	CU	U	North Central Region	S	1982-84=100	JUL1990	1573

The OUTALL= data set in Output 12.2.3 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;
```

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 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 12.2.3).

- You can use PROC FORMAT to define formats for series or BY variables to enhance your output. Note that PROC DATASOURCE associates 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 12.2.3).

Example 12.3: 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 real estate and construction industries with respect to states from March 1989 to March 1990.

The State and Area, Employment, Hours and Earnings survey designates the totals for statewide figures 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.

Output 12.3.1 and Output 12.3.2 show statewide figures for total employment (SA1) in many industries from March 1989 through March 1990.

```
filename ascifile "%sysget(DATASRC_DATA)blseesa.dat" RECFM=F LRECL=152;
proc datasource  filetype=blseesa
                 infile=ascifile
                 outall=totkey
                 out=totemp;
   keep sa1;
   range from 1989:3 to 1990:3;
   rename sa1=totemp;
run;

title1 'Information on Total Employment, OUTALL= Data Set';
proc print data=totkey;
run;

title1 'Total Employment, OUT= Data Set';
proc print data=totemp;
run;
```

Output 12.3.1 Printout of the OUTALL= Data Set for All BY Groups

Information on Total Employment, OUTALL= Data Set

Obs	STATE	AREA	DIVISION	INDUSTRY	DETAIL	NAME	KEPT	SELECTED	TYPE	LENGTH	VARNUM	BLKNUM
1	5	2580	7	0000	1	totemp	1	1	1	5	7	3
2	6	0360	4	2039	6	totemp	1	1	1	5	7	6
3	6	6000	4	2300	2	totemp	1	1	1	5	7	7
4	6	7120	2	0000	1	totemp	1	1	1	5	7	8
5	10	0000	7	6102	6	totemp	1	1	1	5	7	10
6	11	8840	6	5600	2	totemp	1	1	1	5	7	11

Obs	LABEL	FORMAT	FORMATL	FORMATD	ST_DATE	END_DATE	NTIME	NOBS	NINRANGE	STATEABB
1	ALL EMP		0	0	JAN1970	JUN1990	246	246	13	AR
2	ALL EMP		0	0	JAN1972	JUN1990	222	222	13	CA
3	ALL EMP		0	0	JAN1972	JUN1990	222	222	13	CA
4	ALL EMP		0	0	JAN1957	DEC1987	372	372	0	CA
5	ALL EMP		0	0	JAN1984	DEC1987	48	48	0	DE
6	ALL EMP		0	0	JAN1972	JUN1990	222	222	13	DC

Obs	AREANAME	INDTITLE	S_CODE	SEASON	UNITS	NDEC
1	FAYETTEVILLE-SPRINGDALE	FINANCE, INSURANCE, AND REAL ESTATE	SAU0525807000011	U		1
2	ANAHEIM-SANTA ANA	CANNED, CURED, AND FROZEN FOODS	SAU0603604203961	U		1
3	OXNARD-VENTURA	APPAREL AND OTHER TEXTILE PRODUCTS	SAU0660004230021	U		1
4	SALINAS-SEASIDE-MONTEREY	CONSTRUCTION	SAU0671202000011	U		1
5	DELAWARE	NONDEPOS. INSTNS. & SEC. & COM. BRKRS.	SAU1000007610261	U		1
6	WASHINGTON MSA	APPAREL AND ACCESSORY STORES	SAU1188406560021	U		1

```
filename datafile "%sysget(DATASRC_DATA)blseesa.dat" RECFM=F LRECL=152;
proc datasource  filetype=blseesa
                 outall=totkey
                 out=totemp;
   where industry='0000';
   keep sa1;
   range from 1989:3 to 1990:3;
   rename sa1=totemp;
run;

title1 'Total Employment for Real Estate and Construction, OUT= Data Set';
proc print data=totemp;
run;
```

Output 12.3.2 Printout of the OUT= Data Set for INDUSTRY=0000

Total Employment for Real Estate and Construction, OUT= Data Set

Obs	STATE	AREA	DIVISION	INDUSTRY	DETAIL	DATE	totemp
1	5	2580	7	0000	1	MAR1989	16
2	5	2580	7	0000	1	APR1989	16
3	5	2580	7	0000	1	MAY1989	16
4	5	2580	7	0000	1	JUN1989	16
5	5	2580	7	0000	1	JUL1989	16
6	5	2580	7	0000	1	AUG1989	16
7	5	2580	7	0000	1	SEP1989	16
8	5	2580	7	0000	1	OCT1989	16
9	5	2580	7	0000	1	NOV1989	16
10	5	2580	7	0000	1	DEC1989	16
11	5	2580	7	0000	1	JAN1990	15
12	5	2580	7	0000	1	FEB1990	15
13	5	2580	7	0000	1	MAR1990	15

Note the following for this example:

- When the INFILE= option is omitted, the fileref assigned to the BLSEESA file is the default value DATAFILE.
- The FROM and TO values in the RANGE statement correspond to monthly data points since the INTERVAL= option defaults to MONTH for the BLSEESA filetype.

Example 12.4: DRI/McGraw-Hill Format CITIBASE Files

Output 12.4.1 and Output 12.4.2 illustrate how to extract weekly series from a sample CITIBASE file. They also demonstrate how the OUTSELECT= option affects the contents of the auxiliary data sets.

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

```
options yearcutoff=1920;

filename datafile "%sysget(DATASRC_DATA)citidem.dat" RECFM=D LRECL=80;

proc datasource filetype=citibase interval=week
                outall=citiall outby=citikey;
run;

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

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

Output 12.4.1 Listing of the OUTBY= CITIKEY Data Set

Summary Information on Weekly Data for CITIDEMO File

Obs	ST_DATE	END_DATE	NTIME	NOBS	NSERIES	NSELECT
1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	6	6

Output 12.4.2 Listing of the OUTALL= CITIALL Data Set

Weekly Series Available in CITIDEMO File

Obs	NAME	SELECTED	TYPE	LENGTH	VARNUM	BLKNUM	FORMAT	FORMATL
1	FF142B	1	1	5	.	36		0
2	WSPCA	1	1	5	.	37		0
3	WSPUA	1	1	5	.	38		0
4	WSPIA	1	1	5	.	39		0
5	WSPGLT	1	1	5	.	40		0
6	FCPOIL	1	1	5	.	41		0

Obs	FORMATD	ST_DATE	END_DATE	NTIME	NOBS	CODE	ATTRIBUT	NDEC
1	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	FF142B	1	2
2	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	WSPCA	1	2
3	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	WSPUA	1	2
4	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	WSPIA	1	2
5	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	WSPGLT	1	2
6	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	FCPOIL	1	4

Note the following from Output 12.4.2:

- The OUTALL= data set reports the time ranges of variables.
- There are six 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.

Output 12.4.3 and Output 12.4.4 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.

```
filename citidemo "%sysget(DATASRC_DATA)citidem.dat" RECFM=D LRECL=80;

proc datasource filetype=citibase infile=citidemo interval=week
                outall=alloff outby=keyoff outselect=off;
   keep WSP:;
run;

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

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

Output 12.4.3 Listing of the OUTBY= Data Set with OUTSELECT=OFF

Summary Information on Weekly Data for CITIDEMO File

Obs	ST_DATE	END_DATE	NTIME	NOBS	NSERIES	NSELECT
1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	6	4

Output 12.4.4 Listing of the OUTALL= Data Set with OUTSELECT=OFF

Weekly Series Available in CITIDEMO File

Obs	NAME	KEPT	SELECTED	ST_DATE	END_DATE	NTIME	NOBS
1	FF142B	0	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
2	WSPCA	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
3	WSPUA	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
4	WSPIA	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
5	WSPGLT	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
6	FCPOIL	0	0	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271

Setting the OUTSELECT= option ON gives results shown in Output 12.4.5 and Output 12.4.6.

```
filename citidemo "%sysget(DATASRC_DATA)citidem.dat" RECFM=D LRECL=80;
proc datasource filetype=citibase infile=citidemo
                interval=week
                outall=allon outby=keyon outselect=on;
   keep WSP:;
run;

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

title1 'Weekly Series Available in CITIDEMO File';
```

```
proc print data=allon( keep=name kept selected st_date
                            end_date ntime nobs );
run;
```

Output 12.4.5 Listing of the OUTBY= Data Set with OUTSELECT=ON

Summary Information on Weekly Data for CITIDEMO File

Obs	ST_DATE	END_DATE	NTIME	NOBS	NSERIES	NSELECT
1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	6	4

Output 12.4.6 Listing of the OUTALL= Data Set with OUTSELECT=ON

Weekly Series Available in CITIDEMO File

Obs	NAME	KEPT	SELECTED	ST_DATE	END_DATE	NTIME	NOBS
1	WSPCA	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
2	WSPUA	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
3	WSPIA	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271
4	WSPGLT	1	1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271

Comparison of Output 12.4.4 and Output 12.4.6 reveals the following:

- The OUTALL= data set contains six (NSERIES) observations when OUTSELECT=OFF, and four (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, but only computed over the selected variables for OUTSELECT=ON. 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.

The KEEP statement in the last two examples illustrates the use of an additional variable, KEPT, in the OUTALL= data sets of Output 12.4.4 and Output 12.4.6. KEPT, which reports the outcome of the KEEP statement, is only added to the OUTALL= data set when there is KEEP statement.

Adding the RANGE statement to the last example generates the data sets in Output 12.4.7 and Output 12.4.8:

```
filename citidemo "%sysget(DATASRC_DATA)citidem.dat" RECFM=D LRECL=80;
proc datasource filetype=citibase infile=citidemo interval=week
                outby=keyrange out=citiout outselect=on;
   keep WSP:;
   range from '01dec1990'd;
run;

title1 'Summary Information on Weekly Data for CITIDEMO File';
proc print data=keyrange;
```

```
run;

title1 'Weekly Data in CITIDEMO File';
proc print data=citiout;
run;
```

Output 12.4.7 Listing of the OUTBY=KEYRANGE Data Set for FILETYPE=CITIBASE

Summary Information on Weekly Data for CITIDEMO File

Obs	ST_DATE	END_DATE	NTIME	NOBS	NINRANGE	NSERIES	NSELECT
1	Sun, 29 Dec 1985	Sun, 3 Mar 1991	271	271	15	6	4

Output 12.4.8 Printout of the OUT=CITIOUT Data Set for FILETYPE=CITIBASE

Weekly Data in CITIDEMO File

Obs	DATE	WSPCA	WSPUA	WSPIA	WSPGLT
1	Sun, 25 Nov 1990	9.77000	9.66000	9.87000	8.62000
2	Sun, 2 Dec 1990	9.75000	9.64000	9.85000	8.47000
3	Sun, 9 Dec 1990	9.59000	9.48000	9.69000	8.22000
4	Sun, 16 Dec 1990	9.62000	9.51000	9.72000	8.35000
5	Sun, 23 Dec 1990	9.70000	9.60000	9.80000	8.48000
6	Sun, 30 Dec 1990	9.64000	9.53000	9.75000	8.31000
7	Sun, 6 Jan 1991	9.70000	9.59000	9.81000	8.62000
8	Sun, 13 Jan 1991	9.80000	9.70000	9.89000	8.58000
9	Sun, 20 Jan 1991	9.66000	9.57000	9.75000	8.36000
10	Sun, 27 Jan 1991	9.65000	9.56000	9.74000	8.38000
11	Sun, 3 Feb 1991	9.52000	9.43000	9.61000	8.16000
12	Sun, 10 Feb 1991	9.38000	9.29000	9.48000	8.14000
13	Sun, 17 Feb 1991	9.38000	9.29000	9.48000	8.21000
14	Sun, 24 Feb 1991	9.61000	9.53000	9.68000	8.50000
15	Sun, 3 Mar 1991	9.61000	9.53000	9.68000	8.50000

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, 15, 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.

Example 12.5: DRI Data Delivery Service Database

This example demonstrates the DRIDDS filetype for the daily Federal Reserve Series fxrates_dds. Use VALIDVARNAME=ANY in your SAS options statement to allow special characters such as @, $, and % to be in the series name. Note the use of long variable names in the OUT= data set in Output 12.5.2 and long labels in the OUTCONT= data set in Output 12.5.1.

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

```
options validvarname=any;
filename datafile "%sysget(DATASRC_DATA)drifxrat.dat" RECFM=F LRECL=80;
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;
   keep rx: ;
   range from '01jan97'd to '31dec99'd;
   format disttek distekfm.;
   format convtek convtkfm.;
run;

title1 'CONTENTS of FXRATES_.DDS File, KEEP RX:';
```

```
proc print
   data=fixrcnt;
run;

title1 'Daily Series Available in FXRATES_.DDS File, KEEP RX:';
proc print
   data=fixr;
run;
```

Output 12.5.1 Listing of the OUTCONT=FIXRCNT Data Set for FILETYPE=DRIDDS

CONTENTS of FXRATES_.DDS File, KEEP RX:

Obs	NAME	KEPT	SELECTED	TYPE	LENGTH	VARNUM	LABEL	FORMAT	FORMATL	FORMATD
1	RXA$%US$@AU	1	1	1	5	2	EXCHANGE RATE IN AUSTRALIAN DOLLAR PER US DOLLAR - AUSTRALIA		0	0
2	RXBF%US$@BE	1	1	1	5	3	EXCHANGE RATE IN BELGIAN FRANCS PER US DOLLAR - BELGIUM		0	0
3	RXDK%US$@DK	1	1	1	5	4	EXCHANGE RATE IN DANISH KRONE PER 100 US DOLLAR - DENMARK		0	0

Obs	SOURCEID	DISTTEK	CONVTEK	STATUS	UPDATE	UPTIME
1	@FACS/DATA.D	Unspecified	Unspecified	0	31JAN97	132605
2	@FACS/DATA.D	Unspecified	Unspecified	0	31JAN97	132544
3	@FACS/DATA.D	Unspecified	Unspecified	0	31JAN97	132544

Output 12.5.2 Printout of the OUT=FIXR Data Set for FILETYPE=DRIDDS

Daily Series Available in FXRATES_.DDS File, KEEP RX:

Obs	DATE	RXA$%US$@AU	RXBF%US$@BE	RXDK%US$@DK
1	01JAN1997	1.26133	31.9200	5.92877
2	02JAN1997	1.26133	31.9200	5.92877
3	03JAN1997	1.26133	31.9200	5.92877
4	04JAN1997	1.27708	32.4620	6.01098
5	05JAN1997	1.27708	32.4620	6.01098
6	06JAN1997	1.27708	32.4620	6.01098
7	07JAN1997	1.27708	32.4620	6.01098
8	08JAN1997	1.27708	32.4620	6.01098
9	09JAN1997	1.27708	32.4620	6.01098
10	10JAN1997	1.27708	32.4620	6.01098
11	11JAN1997	1.28443	32.9360	6.09112
12	12JAN1997	1.28443	32.9360	6.09112
13	13JAN1997	1.28443	32.9360	6.09112
14	14JAN1997	1.28443	32.9360	6.09112
15	15JAN1997	1.28443	32.9360	6.09112
16	16JAN1997	1.28443	32.9360	6.09112
17	17JAN1997	1.28443	32.9360	6.09112
18	18JAN1997	1.29195	33.7500	6.24658
19	19JAN1997	1.29195	33.7500	6.24658
20	20JAN1997	1.29195	33.7500	6.24658
21	21JAN1997	1.29195	33.7500	6.24658
22	22JAN1997	1.29195	33.7500	6.24658
23	23JAN1997	1.29195	33.7500	6.24658
24	24JAN1997	1.29195	33.7500	6.24658
25	25JAN1997	1.30133	33.8974	6.27520
26	26JAN1997	1.30133	33.8974	6.27520
27	27JAN1997	1.30133	33.8974	6.27520
28	28JAN1997	1.30133	33.8974	6.27520
29	29JAN1997	1.30133	33.8974	6.27520
30	30JAN1997	1.30133	33.8974	6.27520
31	31JAN1997	1.30133	33.8974	6.27520

Example 12.6: PC Format CITIBASE Database

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

Population estimate series for all ages of females including those in the armed forces overseas are given by PANF, while PANM gives the population estimate for all ages of males including those in armed forces overseas. More population estimate time series are described in Output 12.6.1 and are output in Output 12.6.2.

The following statements extract the required population estimates series:

```
filename keyfile "%sysget(DATASRC_DATA)basekey.dat" RECFM=V LRECL=22;
filename indfile "%sysget(DATASRC_DATA)baseind.dat" RECFM=F LRECL=84;
filename dbfile "%sysget(DATASRC_DATA)basedb.dat" RECFM=F LRECL=4;

proc datasource filetype=citidisk  infile=( keyfile indfile dbfile )
                out=popest outall=popinfo;

run;

proc print data=popinfo;
run;
proc print data=popest;
run;
```

Output 12.6.1 Listing of the OUTALL=POPINFO Data Set for FILETYPE=CITIDISK

Daily Series Available in FXRATES_.DDS File, KEEP RX:

Obs	NAME	SELECTED	TYPE	LENGTH	VARNUM	BLKNUM	LABEL	FORMAT	FORMATL
1	PAN	1	1	5	2	1	POPULATION EST.: ALL AGES, INC.ARMED F. OVERSEAS(THOUS.,ANNUAL)		0
2	PAN17	1	1	5	3	2	POPULATION EST.: 16 YRS AND OVER,INC ARMED F.OVERSEAS(THOUS,ANNUAL)		0
3	PAN18	1	1	5	4	3	POPULATION EST.: 18-64 YRS,INC.ARMED F.OVERSEAS(THOUS,ANNUAL)		0
4	PANF	1	1	5	5	4	POPULATION EST.: FEMALES,ALL AGES,INC.ARMED F.O'SEAS(THOUS.,ANN)		0
5	PANM	1	1	5	6	5	POPULATION EST.: MALES, ALL AGES, INC.ARMED F.O'SEAS(THOUS.,ANN)		0

Obs	FORMATD	ST_DATE	END_DATE	NTIME	NOBS	DISKNUM	ATTRIBUT	NDEC	AGGREGAT
1	0	1980	1989	10	10	1	1	0	0
2	0	1980	1989	10	10	1	1	0	0
3	0	1980	1989	10	10	1	1	0	0
4	0	1980	1989	10	10	1	1	0	0
5	0	1980	1989	10	10	1	1	0	0

Output 12.6.2 Printout of the OUT=POPEST Data Set for FILETYPE=CITIDISK

Daily Series Available in FXRATES_.DDS File, KEEP RX:

Obs	DATE	PAN	PAN17	PAN18	PANF	PANM
1	1980	227757	172456	138358	116869	110888
2	1981	230138	175017	140618	118074	112064
3	1982	232520	177346	142740	119275	113245
4	1983	234799	179480	144591	120414	114385
5	1984	237001	181514	146257	121507	115494
6	1985	239279	183583	147759	122631	116648
7	1986	241625	185766	149149	123795	117830
8	1987	243942	187988	150542	124945	118997
9	1988	246307	189867	152113	126118	120189
10	1989	248762	191570	153695	127317	121445

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 12.7: 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 companies 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 "%sysget(DATASRC_DATA)csqibm.dat" recfm=s370v
   lrecl=4820 blksize=14476;
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';
   range  from 1990:4;

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 Last Quarter of 1990';
proc print data=stocks ;
run;
```

Output 12.7.1 contains a listing of the STOCKS data set.

Output 12.7.1 Listing of the OUT=STOCKS Data Set

Common Stocks for Last Quarter of 1990

Obs	DNUM	CNUM	CIC	FILE	EIN	STK	SMBL	ZLIST	XREL	FIC
1	2670	293308	102	6	56-0481457	0	ENGH	6	0	0
2	2835	372917	104	6	06-1047163	0	GENZ	6	0	0
3	3564	896726	106	6	25-0922753	0	TRON	6	0	0
4	3576	172755	100	6	77-0024818	0	CRUS	6	0	0
5	3577	602191	108	6	11-2693062	0	MILT	6	0	0
6	3630	616350	104	6	34-0299600	0	MORF	6	0	0
7	3674	827079	203	6	94-1527868	0	SILI	6	0	0
8	3842	602720	104	6	25-0668780	0	MNES	6	0	0
9	5080	007698	103	6	59-1001822	0	AESM	6	0	0
10	5122	090324	104	6	84-0601662	0	BIND	6	0	0
11	5211	977865	104	6	38-1746752	0	WLHN	6	0	0
12	5600	299155	101	6	36-1050870	0	EVAN	6	0	0
13	5731	382091	106	6	94-2366177	0	GGUY	6	0	0
14	7372	45812M	104	6	94-2658153	0	INTS	6	0	0
15	7372	566140	109	6	04-2711580	0	MCAM	6	0	0
16	7373	913077	103	6	81-0422894	0	TOTE	6	0	0
17	7510	008450	108	6	34-1050582	0	AGNC	6	0	0
18	7819	026038	307	6	23-2359277	0	AFTI	6	0	0
19	8700	055383	103	6	59-1781257	0	BEIH	6	0	0
20	8731	759916	109	6	04-2729386	0	RGEN	6	0	0

Obs	INCORP	STATE	COUNTY	DATE	comstock	ftcomstk	CONAME
1	10	13	121	1990:4	16.2510		ENGRAPH INC
2	10	25	17	1990:4	0.1620		GENZYME CORP
3	42	37	105	1990:4	3.1380		TRION INC
4	6	6	85	1990:4	.		CIRRUS LOGIC INC
5	10	36	103	1990:4	.		MILTOPE GROUP INC
6	39	39	35	1990:4	.		MOR-FLO INDS
7	10	6	85	1990:4	.		SILICONIX INC
8	42	42	3	1990:4	6.7540		MINE SAFETY APPLIANCES CO
9	12	12	25	1990:4	.		AERO SYSTEMS INC
10	18	18	97	1990:4	3.2660		BINDLEY WESTERN INDS
11	26	26	145	1990:4	6.4800		WOLOHAN LUMBER CO
12	10	17	31	1990:4	.		EVANS INC
13	6	6	75	1990:4	0.0520		GOOD GUYS INC
14	6	6	85	1990:4	.		INTEGRATED SYSTEMS INC
15	25	25	17	1990:4	0.0770		MARCAM CORPORATION
16	10	30	111	1990:4	0.0570		UNITED TOTE INC
17	10	39	35	1990:4	.		AGENCY RENT-A-CAR INC
18	10	42	45	1990:4	0.0210		AMERICAN FILM TECHNOL
19	10	13	121	1990:4	0.5170		BEI HOLDINGS LTD
20	10	25	17	1990:4	.		REPLIGEN 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 FILETYPE=CS48QUC option instead.

Example 12.8: Annual COMPUSTAT Data Files, V9.2 New Filetype CSAUC3

Annual COMPUSTAT data in Universal Character format is read for PRICES since the year 2002, so that the desired output show the PRICE (HIGH), PRICE (LOW), and PRICE (CLOSE) for each company.

```
filename datafile "%sysget(DATASRC_DATA)csaucy3.dat" RECFM=F LRECL=13612;
/*-------------------------------------------------------------*
 * create OUT=csauy3 data set with ASCII 2003 Industrial Data  *
 * compare it with the OUT=csauc data set created by DATA STEP *
 *-------------------------------------------------------------*/

proc datasource filetype=csaucy3 ascii
                infile=datafile
                interval=year
                outselect=on
                outkey=y3key
                out=csauy3;

     keep data197-data199 label;
     range from 2002;
run;

proc sort
   data=csauy3 out=csauy3;
   by dnum cnum cic file zlist smbl xrel stk;
run;

title1 'Price, High, Low and Close for Range from 2002';
proc contents data=csauy3;
run;

proc print data=csauy3;
run;
```

Output 12.8.1 shows information on the contents of the CSAUY3 data set while Output 12.8.2 shows a listing of the CSAUY3 data set.

Output 12.8.1 Listing of the CONTENTS of OUT=CSAUY3 Data Set

Price, High, Low and Close for Range from 2002

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
3	CIC	Char	3		
2	CNUM	Char	6		
11	COUNTY	Num	5		
13	CPSPIN	Char	1		
15	CSSPII	Char	1		
14	CSSPIN	Char	2		
18	DATA197	Num	5		Price - Fiscal Year - High ($&c,NA)
19	DATA198	Num	5		Price - Fiscal Year - Low ($&c,NA)
20	DATA199	Num	5		Price - Close - Fiscal Year-End ($&c,NA)
17	DATE	Num	4	YEAR4.	Date of Observation
1	DNUM	Num	5		
9	DUPFILE	Num	5		
16	EIN	Char	10		
4	FILE	Num	5		
12	FINC	Num	5		
6	SMBL	Char	8		
10	STATE	Num	5		
8	STK	Num	5		
7	XREL	Num	5		
5	ZLIST	Num	5		

Output 12.8.2 Listing of the OUT=CSAUY3 Data Set

Price, High, Low and Close for Range from 2002

Obs	DNUM	CNUM	CIC	FILE	ZLIST	SMBL	XREL	STK	DUPFILE	STATE	COUNTY	FINC	CPSPIN
1	3089	899896	104	11	1	TUP	444	0	0	12	95	0	1
2	3089	899896	104	11	1	TUP	444	0	0	12	95	0	1
3	3674	032654	105	11	1	ADI	928	0	0	25	21	0	1
4	3674	032654	105	11	1	ADI	928	0	0	25	21	0	1
5	3842	053801	106	1	5	AVR	0	0	0	25	21	0	
6	3842	053801	106	1	5	AVR	0	0	0	25	21	0	
7	6035	149547	101	3	25	CAVB	0	0	0	47	149	0	
8	6035	149547	101	3	25	CAVB	0	0	0	47	149	0	
9	6211	617446	448	11	1	MWD	725	0	0	36	61	0	1
10	6211	617446	448	11	1	MWD	725	0	0	36	61	0	1
11	6726	09247M	105	1	4	BMN	0	0	0	34	13	0	
12	6726	09247M	105	1	4	BMN	0	0	0	34	13	0	
13	7011	54021P	205	1	5	LGN	0	0	0	13	121	0	
14	7011	54021P	205	1	5	LGN	0	0	0	13	121	0	
15	7370	35921T	108	1	5	FNT	0	0	0	36	87	0	
16	7370	35921T	108	1	5	FNT	0	0	0	36	87	0	
17	7370	459200	101	11	1	IBM	903	0	0	36	119	0	1
18	7370	459200	101	11	1	IBM	903	0	0	36	119	0	1
19	7812	591610	100	1	4	MGM	0	0	0	6	37	0	
20	7812	591610	100	1	4	MGM	0	0	0	6	37	0	

Obs	CSSPIN	CSSPII	EIN	DATE	DATA197	DATA198	DATA199
1	10		36-4062333	2002	24.990	14.4000	15.0800
2	10		36-4062333	2003	.	.	.
3	10		04-2348234	2002	48.840	17.8800	26.8000
4	10		04-2348234	2003	.	.	.
5			06-1174053	2002	1.500	0.2200	0.2300
6			06-1174053	2003	.	.	.
7			62-1721072	2002	14.000	11.5810	13.3400
8			62-1721072	2003	.	.	.
9	10	1	36-3145972	2002	60.020	28.8010	45.2400
10	10	1	36-3145972	2003	.	.	.
11				2002	11.050	10.3700	11.0100
12				2003	.	.	.
13			52-2093696	2002	13.894	1.0084	13.8940
14			52-2093696	2003	.	.	.
15			13-3950283	2002	0.440	0.1200	0.2600
16			13-3950283	2003	.	.	.
17	10	1	13-0871985	2002	126.390	54.0100	77.5000
18	10	1	13-0871985	2003	.	.	.
19			95-4605850	2002	23.250	9.0000	13.0000
20			95-4605850	2003	.	.	.

Note that annual COMPUSTAT data are available in either IBM 360/370 General format or 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 12.9: CRSP Daily NYSE/AMEX Combined Stocks

This sample code 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 that this database comprises:

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 in 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 in which you want to read them. 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 in which they are specified. Therefore, you can request that all three volumes be mounted to the same drive, if you choose to do so.

This sample code illustrates the following points:

- 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 which 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 12.4. 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;
```

The KEEP statement 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 sample code outputs only NCUSIP and SHROUT to the OUTEVENT= data set for CUSIP='09523220':

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

Output 12.9.1, Output 12.9.2, Output 12.9.3, and Output 12.9.4 show how to read the CRSP Daily NYSE/AMEX Combined ASCII Character Files.

```
filename dxci "%sysget(DATASRC_DATA)dxccal95.dat" RECFM=F LRECL=130;
filename dxc "%sysget(DATASRC_DATA)dxcsub95.dat" RECFM=F LRECL=400;

/*--- create output data sets from character format DX files ---*/
/*- create securities output data sets using DATASOURCE  -------*/
/*- statements                                                 -*/
proc datasource filetype=crspdcs  ascii
                infile=( dxci dxc )
                interval=day
                outcont=dxccont
                outkey=dxckey
                outall=dxcall
                out=dxc
```

```
                outevent=dxcevent
                outselect=off;
   range from '15aug95'd to '28aug95'd ;
   where cusip in ('12709510','35614220');
run;

title1 'Date Range 15aug95-28aug95 ';

title3 'DX Security File Outputs';
title4 'OUTKEY= Data Set';
proc print data=dxckey;
run;

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

title4 "Listing of OUT= Data Set for cusip in ('12709510','35614220')";
proc print data=dxc;
run;

title4 "Listing of OUTEVENT= Data Set for cusip in ('12709510','35614220')";
proc print data=dxcevent;
run;
```

Output 12.9.1 Listing of the OUTBY= Data Set with OUTSELECT=OFF

Date Range 15aug95-28aug95

DX Security File Outputs
OUTKEY= Data Set

Obs	CUSIP	PERMNO	COMPNO	ISSUNO	HEXCD	HSICCD	BYSELECT	ST_DATE	END_DATE
1	68391610	10000	7952	9787	3	3990	0	07JAN1986	11JUN1987
2	12709510	10010	7967	9809	3	3840	1	17JAN1986	28AUG1995
3	49307510	10020	7972	9824	3	6710	0	27JAN1986	30APR1993
4	00338690	10030	22160	0	1	3310	0	02JUL1962	26DEC1968
5	41741F20	10040	7988	9846	3	6210	0	07FEB1986	15JUN1989
6	00074210	10050	13	11	3	3448	0	29DEC1972	16JUN1978
7	35614220	10060	8007	9876	3	1040	1	24FEB1986	29DEC1995

Obs	NTIME	NOBS	NINRANGE	NSERIES	NSELECT
1	521	0	0	35	7
2	3511	2431	10	35	7
3	2651	0	0	35	7
4	2370	0	0	35	7
5	1225	0	0	35	7
6	1996	0	0	35	7
7	3596	2492	10	35	7

Output 12.9.2 Listing of the OUTCONT= Data Set

Date Range 15aug95-28aug95

DX Security File Outputs
OUTCONT= Data Set

Obs	NAME	KEPT	SELECTED	TYPE	LENGTH	VARNUM	LABEL	FORMAT	FORMATL	FORMATD
1	BIDLO	1	1	1	6	8	Bid or Low		0	0
2	ASKHI	1	1	1	6	9	Ask or High		0	0
3	PRC	1	1	1	6	10	Closing Price of Bid/Ask average		0	0
4	VOL	1	1	1	6	11	Share Volume		0	0
5	RET	1	1	1	6	12	Holding Period Return		0	0
6	SXRET	1	1	1	6	13	Standard Deviation Excess Return		0	0
7	BXRET	1	1	1	6	14	Beta Excess Return		0	0
8	NCUSIP	0	0	2	8	.	Name CUSIP		0	0
9	TICKER	0	0	2	5	.	Exchange Ticker Symbol		0	0
10	COMNAM	0	0	2	32	.	Company Name		0	0
11	SHRCLS	0	0	2	1	.	Share Class		0	0
12	SHRCD	0	0	1	6	.	Share Code		0	0
13	EXCHCD	0	0	1	6	.	Exchange Code		0	0
14	SICCD	0	0	1	6	.	Standard Industrial Classification Code		0	0
15	DISTCD	0	0	1	6	.	Distribution Code		0	0
16	DIVAMT	0	0	1	6	.	Dividend Cash Amount		0	0
17	FACPR	0	0	1	6	.	Factor to adjust price		0	0
18	FACSHR	0	0	1	6	.	Factor to adjust shares outstanding		0	0
19	DCLRDT	0	0	1	6	.	Declaration date	DATE	7	0
20	RCRDDT	0	0	1	6	.	Record date	DATE	7	0
21	PAYDT	0	0	1	6	.	Payment date	DATE	7	0
22	SHROUT	0	0	1	6	.	Number of shares outstanding		0	0
23	SHRFLG	0	0	1	6	.	Share flag		0	0
24	DLSTCD	0	0	1	6	.	Delisting code		0	0
25	NWPERM	0	0	1	6	.	New CRSP permanent number		0	0
26	NEXTDT	0	0	1	6	.	Date of next available information	DATE	7	0
27	DLBID	0	0	1	6	.	Delisting bid		0	0
28	DLASK	0	0	1	6	.	Delisting ask		0	0
29	DLPRC	0	0	1	6	.	Delisting price		0	0
30	DLVOL	0	0	1	6	.	Delisting volume		0	0
31	DLRET	0	0	1	6	.	Delisting return		0	0
32	TRTSCD	0	0	1	6	.	Traits code		0	0
33	NMSIND	0	0	1	6	.	National Market System Indicator		0	0
34	MMCNT	0	0	1	6	.	Market maker count		0	0
35	NSDINX	0	0	1	6	.	NASD index		0	0

Output 12.9.3 Listing of the OUT= Data Set with OUTSELECT=OFF for CUSIPs 12709510 and 35614220

Date Range 15aug95-28aug95

DX Security File Outputs
Listing of OUT= Data Set for cusip in ('12709510','35614220')

Obs	CUSIP	PERMNO	COMPNO	ISSUNO	HEXCD	HSICCD	DATE	BIDLO	ASKHI
1	12709510	10010	7967	9809	3	3840	15AUG1995	7.500	7.8750
2	12709510	10010	7967	9809	3	3840	16AUG1995	7.500	7.8750
3	12709510	10010	7967	9809	3	3840	17AUG1995	7.500	7.8750
4	12709510	10010	7967	9809	3	3840	18AUG1995	7.375	7.5000
5	12709510	10010	7967	9809	3	3840	21AUG1995	7.375	7.3750
6	12709510	10010	7967	9809	3	3840	22AUG1995	7.250	7.3750
7	12709510	10010	7967	9809	3	3840	23AUG1995	7.250	7.3750
8	12709510	10010	7967	9809	3	3840	24AUG1995	7.125	7.5000
9	12709510	10010	7967	9809	3	3840	25AUG1995	6.875	7.3750
10	12709510	10010	7967	9809	3	3840	28AUG1995	7.000	7.1250
11	35614220	10060	8007	9876	3	1040	15AUG1995	12.375	12.6875
12	35614220	10060	8007	9876	3	1040	16AUG1995	12.125	12.3750
13	35614220	10060	8007	9876	3	1040	17AUG1995	12.250	12.3125
14	35614220	10060	8007	9876	3	1040	18AUG1995	12.250	12.6250
15	35614220	10060	8007	9876	3	1040	21AUG1995	12.375	12.6250
16	35614220	10060	8007	9876	3	1040	22AUG1995	12.250	12.3750
17	35614220	10060	8007	9876	3	1040	23AUG1995	12.125	12.2500
18	35614220	10060	8007	9876	3	1040	24AUG1995	12.125	12.3750
19	35614220	10060	8007	9876	3	1040	25AUG1995	12.000	12.2500
20	35614220	10060	8007	9876	3	1040	28AUG1995	12.000	12.0625

Obs	PRC	VOL	RET	SXRET	BXRET
1	7.5625	29200	-0.008197	.	.
2	7.5000	22365	-0.008264	.	.
3	7.5000	33416	0.000000	.	.
4	7.3750	16666	-0.016667	.	.
5	7.3750	9382	0.000000	.	.
6	7.2500	33674	-0.016949	.	.
7	7.3125	22371	0.008621	.	.
8	7.1250	38621	-0.025641	.	.
9	7.0000	29713	-0.017544	.	.
10	7.0000	38798	0.000000	.	.
11	12.3750	39136	0.000000	.	.
12	12.2031	45916	-0.013889	.	.
13	12.2500	43644	0.003841	.	.
14	12.3750	11027	0.010204	.	.
15	12.3750	7378	0.000000	.	.
16	12.2500	99655	-0.010101	.	.
17	12.1250	95148	-0.010204	.	.
18	12.3750	185572	0.020619	.	.
19	12.0000	9575	-0.030303	.	.
20	12.0625	12854	0.005208	.	.

Output 12.9.4 Listing of the OUTEVENT= Data Set in Range 15aug95-28aug95

Date Range 15aug95-28aug95

DX Security File Outputs
Listing of OUTEVENT= Data Set for cusip in ('12709510','35614220')

Obs	CUSIP	PERMNO	COMPNO	ISSUNO	HEXCD	HSICCD	EVENT	DATE	NCUSIP	TICKER	COMNAM	SHRCLS
1	12709510	10010	7967	9809	3	3840	DELIST	28AUG1995				
2	12709510	10010	7967	9809	3	3840	NASDIN	24AUG1995				

Obs	SHRCD	EXCHCD	SICCD	DISTCD	DIVAMT	FACPR	FACSHR	DCLRDT	RCRDDT	PAYDT	SHROUT	SHRFLG
1	.	.	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.	.	.

Obs	DLSTCD	NWPERM	NEXTDT	DLBID	DLASK	DLPRC	DLVOL	DLRET	TRTSCD	NMSIND	MMCNT	NSDINX
1	203	23588	.	.	.	0	.	0.037500	.	.	.	.
2	.	.	.	.	.	.	.	.	1	2	17	2

Note in Output 12.9.4 that there were no events in range for cusip 35614220. See Chapter 38, "The SASECRSP Interface Engine," for more on CRSPAccess Data access.

References

Bureau of Economic Analysis (1986), *The National Income and Product Accounts of the United States, 1929–82*, Washington, DC: U.S. Department of Commerce.

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

Bureau of Economic Analysis (1991), *Survey of Current Business*, Washington, DC: U.S. Department of Commerce.

Bureau of Labor Statistics (2012), `http://www.bls.gov/`.

Center for Research in Security Prices (2000), *CRSP SFA Guide*, Chicago: CRSP, Chicago Booth.

Center for Research in Security Prices (2006a), *CRSP Data Description Guide*, Chicago: CRSP, Chicago Booth.

Center for Research in Security Prices (2006b), *CRSP Fortran-77 to Fortran-95 Migration Guide*, Chicago: CRSP, Chicago Booth.

Center for Research in Security Prices (2006c), *CRSP Programmer's Guide*, Chicago: CRSP, Chicago Booth.

Center for Research in Security Prices (2006d), *CRSP Utilities Guide*, Chicago: CRSP, Chicago Booth.

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

Citibank (1991a), *CITIBASE–Daily*, New York: Citibank.

Citibank (1991b), *CITIBASE–Weekly*, New York: Citibank.

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

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

FAME Information Services (1995), *User's Guide to FAME*, Ann Arbor, MI: FAME Information Services.

Haver Analytics (2012), *DLX Database Profiles*, New York: Haver Analytics, accessed November 27, 2012. URL http://www.haver.com/databaseprofiles.html

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

Organization for Economic Cooperation and Development (1992a), *Annual National Accounts, Vol. 1: Main Aggregates Content Documentation for Magnetic Tape Subscription*, Paris: OECD.

Organization for Economic Cooperation and Development (1992b), *Annual National Accounts, Vol. 2: Detailed Tables Technical Documentation for Magnetic Tape Subscription*, Paris: OECD.

Organization for Economic Cooperation and Development (1992c), *Main Economic Indicators Database Note*, Paris: OECD.

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

Organization for Economic Cooperation and Development (1992e), *Quarterly National Accounts Inventory of Series Codes*, Paris: OECD.

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

Standard & Poor's Compustat Services (2003), *Compustat Technical Guide*, Englewood, CO: McGraw-Hill.

Subject Index

Syntax Index