Chapter 5
The SASEFAME Interface Engine

Chapter Table of Contents

Chapter 5
The SASEFAME Interface Engine

Overview

The SASEFAME interface engine allows SAS users the ability to access and process time series data residing in a FAME database, and provides a seamless interface between FAME and SAS data processing.

The SASEFAME engine uses the LIBNAME statement to allow you to specify which time series you would like to read from the FAME database, and how you would like to convert the selected time series to the same time scale. The SAS Data step can then be used to perform further subsetting and to store the resulting time series into a SAS data set. You can perform more analysis if desired either in the same SAS session or in another session at a later time.

Getting Started

Structure of a SAS Data Set Containing Time Series Data

SAS requires 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 SASEFAME engine does provide a time ID variable by the name of DATE.

Reading and Converting FAME Database Time Series

The SASEFAME engine supports reading and converting time series stored in FAME databases. The SASEFAME engine uses the FAME WORK database to temporarily store the converted time series. Only the series specified by the FAME wildcard are written to the FAME WORK database.

Using the SAS DATA step

If desired, you may store the converted series in a SAS data set by using the SAS DATA step. You may also perform other operations on your data inside the DATA step. Once your data is stored in a SAS data set you may use it as you would any other SAS data set.

Using SAS Procedures

You can print the output SAS data set by using the PRINT procedure and report information concerning the contents of your data set by using the CONTENTS procedure. See Example 5.1. You can create a view of the FAME data base by using the SQL procedure to create your view using the SASEFAME engine in your libref, along with the using clause. See Example 5.5.

Syntax

The SASEFAME engine uses standard engine syntax:

LIBNAME *libref* **sasefame** *'physical name' options***;**

Options used by SASEFAME are summarized in the table below.

Description	Statement	Option
specifies the FAME frequency and the FAME technique.	LIBNAME libref SASEFAME	CONVERT=
specifies a FAME wildcard to match data object series names within the FAME database, which limits the selection of time series that are included in the SAS data set.	LIBNAME libref SASEFAME	WILDCARD=

The LIBNAME libref SASEFAME Statement

LIBNAME libref SASEFAME 'physical name' *options;*

The following options can be used in the LIBNAME libref SASEFAME statement:

CONVERT=*(FREQ=fame_frequency TECH=fame_technique)*

specifies the FAME frequency and the FAME technique just as you would in the FAME CONVERT function. There are four possible values for *fame_technique*: CONSTANT (default), CUBIC, DISCRETE, or LINEAR. All FAME frequencies except PPY and YPP are supported by the SASEFAME engine. For all possible *fame_frequency* values, see "Understanding Frequencies" in your *USER'S GUIDE TO FAME*. For a more complete discussion of FAME frequencies and SAS time intervals see *"Mapping FAME Frequencies to SAS Time Intervals"* later in this chapter. As an example,

```
LIBNAME libref sasefame 'physical-name'
   CONVERT=(TECH=CONSTANT FREQ=TWICEMONTHLY);
```

WILDCARD=*"fame_wildcard"*

By default, the SASEFAME engine reads all time series in the FAME database that you name in your SASEFAME libref. You can limit the time series read from the FAME database by specifying the WILDCARD= option on your LIBNAME statement. The *fame_wildcard* is a quoted string containing the FAME wildcard you wish to use. The wildcard is used against the data object names (time series only) in the FAME database that resides in the library you are in the process of assigning. For more information about wildcarding, see "Specifying Wildcards" in your *USER'S GUIDE TO FAME*.

For example, to read all time series in the TEST library being accessed by the SASEFAME engine, you would specify

```
LIBNAME test sasefame 'physical name of test data base'
   WILDCARD="?";
```

To read series with names such as A_DATA, B_DATA, C_DATA, you could specify

```
LIBNAME test sasefame 'physical name of test data base'
   WILDCARD="^_DATA";
```

When you use the WILD= option, you are limiting the number of time series that are read and converted to the desired frequency. This option can help you save resources when processing large databases or when processing a large number of observations, i.e. daily, hourly, minutely frequencies. Since the SASEFAME engine uses the FAME WORK database to store the converted time series, using wildcards is recommended to prevent your WORK space from getting too large.

Details

The SAS Output Data Set

You can use the SAS DATA step to write the FAME converted series to a SAS data set. This allows the user the ability to easily analyze the data using SAS. The name of the output data set is specified by you on the DATA statement. This causes the engine supervisor to create a SAS data set using the specified name in either the SAS WORK library, or if specified, the USER library. For more about naming your SAS data set see "Characteristics of SAS Data Libraries" in *SAS Language: Reference, Version 7*.

The contents of the SAS data set include the DATE of each observation, the series names of each series read from the FAME database as specified by the WILDCARD option, and the label or FAME description of each series. Missing values are represented as '.' in the SAS data set. You can use PROC PRINT and PROC CONTENTS to print your output data set and its contents. You can use PROC SQL along with the SASEFAME engine to create a view of your SAS data set.

The DATE variable in the SAS data set contains the date of the observation. It is important to note that FAME frequencies often have a sample unit in parentheses following the keyword frequency. This sample unit is an end-of-interval unit. SAS dates are represented using begin-of-interval notation. For FAME weekly intervals that end on a Friday, FAME reports the date on the Friday that ends the week whereas SAS reports the date on the Saturday that begins the week. A more detailed discussion of how to map FAME frequencies to SAS Time Intervals follows.

Mapping FAME Frequencies to SAS Time Intervals

The following table summarizes the mapping of Fame frequencies to SAS time intervals. It is important to note that FAME frequencies often have a sample unit in parentheses following the keyword frequency. This sample unit is an end-of-interval unit. SAS dates are represented using begin-of-interval notation. For more on SAS time intervals, see "Date Intervals, Formats, and Functions" in your *SAS/ETS User's Guide*. For more on FAME frequencies, see "Understanding Frequencies" in your *USER'S GUIDE TO FAME*.

FAME FREQUENCY	SAS TIME INTERVAL
WEEKLY (SUNDAY)	WEEK.2
WEEKLY (MONDAY)	WEEK.3
WEEKLY (TUESDAY)	WEEK.4
WEEKLY (WEDNESDAY)	WEEK.5
WEEKLY (THURSDAY)	WEEK.6
WEEKLY (FRIDAY)	WEEK.7
WEEKLY (SATURDAY)	WEEK.1
BIWEEKLY (ASUNDAY)	WEEK2.2
BIWEEKLY (AMONDAY)	WEEK2.3

FAME FREQUENCY	SAS TIME INTERVAL
BIWEEKLY (ATUESDAY)	WEEK2.4
BIWEEKLY (AWEDNESDAY)	WEEK2.5
BIWEEKLY (ATHURSDAY)	WEEK2.6
BIWEEKLY (AFRIDAY)	WEEK2.7
BIWEEKLY (ASATURDAY)	WEEK2.1
BIWEEKLY (BSUNDAY)	WEEK2.9
BIWEEKLY (BMONDAY)	WEEK2.10
BIWEEKLY (BTUESDAY)	WEEK2.11
BIWEEKLY (BWEDNESDAY)	WEEK2.12
BIWEEKLY (BTHURSDAY)	WEEK2.13
BIWEEKLY (BFRIDAY)	WEEK2.14
BIWEEKLY (BSATURDAY)	WEEK2.8
BIMONTHLY (NOVEMBER)	MONTH2.2
BIMONTHLY	MONTH2.1
QUARTERLY (OCTOBER)	QTR.2
QUARTERLY (NOVEMBER)	QTR.3
QUARTERLY	QTR.1
ANNUAL (JANUARY)	YEAR.2
ANNUAL (FEBRUARY)	YEAR.3
ANNUAL (MARCH)	YEAR.4
ANNUAL (APRIL)	YEAR.5
ANNUAL (MAY)	YEAR.6
ANNUAL (JUNE)	YEAR.7
ANNUAL (JULY)	YEAR.8
ANNUAL (AUGUST)	YEAR.9
ANNUAL (SEPTEMBER)	YEAR.10
ANNUAL (OCTOBER)	YEAR.11
ANNUAL (NOVEMBER)	YEAR.12
ANNUAL	YEAR.1
SEMIANNUAL (JULY)	SEMIYEAR.2
SEMIANNUAL (AUGUST)	SEMIYEAR.3
SEMIANNUAL (SEPTEMBER)	SEMIYEAR.4
SEMIANNUAL (OCTOBER)	SEMIYEAR.5
SEMIANNUAL (NOVEMBER)	SEMIYEAR.6
SEMIANNUAL	SEMIYEAR.1
YPP	not supported
PPY	not supported
SECONDLY	SECOND
MINUTELY	MINUTE

FAME FREQUENCY	SAS TIME INTERVAL
HOURLY	HOUR
DAILY	DAY
BUSINESS	WEEKDAY
TENDAY	TENDAY
TWICEMONTHLY	SEMIMONTH
MONTHLY	MONTH

Examples

Example 5.1. Converting an entire FAME database

To enable conversion of all Time Series no wildcard is specified, so the default "?" wildcard is used. Always consider both the number of time series and the number of observations generated by the conversion process. The converted series are stored in the FAME WORK database during the SAS DATA step. You may further limit your resulting SAS data set by using KEEP, DROP, or WHERE statements inside your data step.

```
libname famedir sasefame '.'
                convert=(freq=annual technique=constant);

libname mydir '/mine/data/europe/sas/oecdir';

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   /* do nothing special */
   run;

proc print data=mydir.a; run;
```

In the above example, the FAME database is called oecd1.db and it resides in the famedir directory. The DATA statement names the SAS output data set 'a' which will reside in mydir. All time series in the FAME oecd1.db data base will be converted to an annual frequency and stored in the mydir.a SAS data set. The PROC PRINT statement creates a listing of all of the observations in the mydir.a SAS data set.

Example 5.2. Reading Time Series from the FAME database

Use the FAME wildcard option to limit the number of series converted. For example, suppose you want to read only series starting with "WSPCA". You could use the following code:

```
libname lib1 sasefame '/mine/data/econ_fame/sampdir'
             wildcard="wspca?"
             convert=(technique=constant freq=twicemonthly );

libname lib2 '/mine/data/econ_sas/sampdir';

data lib2.twild(label='Annual Series from the FAMEECON.db');
   set lib1.subecon;
   /* keep only  */
   keep date wspca;
   run;

proc contents data=lib2.twild; run;

proc print data=lib2.twild; run;
```

The wildcard="wspca?" option limits reading only those series whose names begin with WSPCA. The SAS KEEP statement further restricts the SAS data set to include only the series named WSPCA and the DATE variable. The time interval used for the conversion is TWICEMONTHLY.

Example 5.3. Writing Time Series to the SAS data set

You can use the KEEP or DROP statement to include or exclude certain series names from the SAS data set.

```
libname famedir sasefame '.'
        convert=(freq=annual technique=constant);

libname mydir '/mine/data/europe/sas/oecdir';

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   drop ita_dird--jpn_herd tur_dird--usa_herd;
   run;

proc print data=mydir.a; run;
```

You can rename your SAS variables by using the RENAME statement.

```
option validvarname=any;

libname famedir sasefame '.'
        convert=(freq=annual technique=constant);

libname mydir '/mine/data/europe/sas/oecdir';

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   /* keep and rename  */
   keep date ita_dird--jpn_herd tur_dird--usa_herd;
   rename ita_dird='Italy.dirdes'n
          jpn_dird='Japan.dirdes'n
          tur_dird='Turkey.dirdes'n
          usa_dird='UnitedStates.dirdes'n ;
   run;

proc print data=mydir.a; run;
```

Example 5.4. Limiting the time range of data

You may also limit the time range of the data in the SAS data set by using the WHERE statement in the data step to process the time id variable DATE only when it falls in the range you are interested in.

```
libname famedir SASEFAME ’.’
        convert=(freq=annual technique=constant);

libname mydir ’/mine/data/europe/sas/oecdir’;

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   /* where only  */
   where date between ’01jan88’d and ’31dec90’d;
   run;

proc print data=mydir.a; run;
```

All data for 1988, 1989, and 1990 are included in the SAS data set. See the "*SAS Language: Reference, Version 7*" for more information on KEEP, DROP, RENAME and WHERE statements.

Example 5.5. Creating a View using SQL Procedure and SASEFAME

This example creates a view using the SQL procedure’s from and using clauses. See "*SQL Procedure Guide, Version 7*" for details on SQL views.

```
title1 ’famesql5: PROC SQL Dual Embedded Librarys w/ FAME option’;
options validvarname=any;

/* Dual Embedded Library Allocations (With FAME Option) */
/********************************************************/

/* OECD1 Database */
/******************/

title2 ’OECD1: Dual Embedded Library Allocations
                  with FAME Option’;
proc sql;
   create view fameview as
   select date, ’fin.herd’n
   from lib1.oecd1
   using libname lib1 sasefame ’/economic/databases/testdat’
                      convert=(tech=constant freq=annual),
         libname temp ’/usr/local/scratch/mine’
   quit;

title2 ’OECD1: Print of View from Embedded Library
                  with FAME Option’;
proc print data=fameview;
run;
```

Output 5.5.1. Printout of the fameview of OECD data

```
                PROC SQL Dual Embedded Librarys w/ FAME option
            OECD1: Print of View from Embedded Library with Option

                           Obs    DATE    FIN.HERD

                            1     1985     1097.0
                            2     1986     1234.0
                            3     1987     1401.3
                            4     1988     1602.0
                            5     1989     1725.5
                            6     1990     1839.0
                            7     1991          .
```

```
/* SUBECON Database */
/*********************/

title2 'SUBECON: Dual Embedded Library Allocations
                 with FAME Option';
proc sql;
   create view fameview as
   select date, gaa
   from lib1.subecon
   using libname lib1 sasefame '/economic/databases/testdat'
                      convert=(tech=constant freq=annual),
         libname temp '/usr/local/scratch/mine'
   quit;

title2 'SUBECON: Print of View from Embedded Library
                      with FAME Option';
proc print data=fameview;
run;
```

Output 5.5.2. Printout of the fameview of DRI Basic Ecomonic data

```
          PROC SQL Dual Embedded Librarys w/ FAME option
     SUBECON: Print of View from Embedded Library with Option

                    Obs    DATE       GAA

                      1    1946          .
                      2    1947          .
                      3    1948     23174.00
                      4    1949     19003.00
                      5    1950     24960.00
                      6    1951     21906.00
                      7    1952     20246.00
                      8    1953     20912.00
                      9    1954     21056.00
                     10    1955     27168.00
                     11    1956     27638.00
                     12    1957     26723.00
                     13    1958     22929.00
                     14    1959     29729.00
                     15    1960     28444.00
                     16    1961     28226.00
                     17    1962     32396.00
                     18    1963     34932.00
                     19    1964     40024.00
                     20    1965     47941.00
                     21    1966     51429.00
                     22    1967     49164.00
                     23    1968     51208.00
                     24    1969     49371.00
                     25    1970     44034.00
                     26    1971     52352.00
                     27    1972     62644.00
                     28    1973     81645.00
                     29    1974     91028.00
                     30    1975     89494.00
                     31    1976    109492.00
                     32    1977    130260.00
                     33    1978    154357.00
                     34    1979    173428.00
                     35    1980    156096.00
                     36    1981    147765.00
                     37    1982    113216.00
                     38    1983    133495.00
                     39    1984    146448.00
                     40    1985    128521.99
                     41    1986    111337.99
                     42    1987    160785.00
                     43    1988    210532.00
                     44    1989    201637.00
                     45    1990    218702.00
                     46    1991    210666.00
                     47    1992          .
                     48    1993          .
```

```
title2 'DB77: Dual Embedded Library Allocations
                 with FAME Option';
proc sql;
   create view fameview as
   select date, ann, 'qandom.x'n
   from lib1.db77
   using libname lib1 sasefame '/economic/databases/testdat'
                      convert=(tech=constant freq=annual),
       libname temp '/usr/local/scratch/mine'
 quit;

title2 'DB77: Print of View from Embedded Library
                 with FAME Option';
proc print data=fameview;
run;
```

Output 5.5.3. Printout of the fameview of DB77 data

```
    PROC SQL Dual Embedded Librarys w/ FAME option
DB77: Print of View from Embedded Library with Option

            Obs    DATE    ANN    QANDOM.X

              1    1959      .     0.56147
              2    1960      .     0.51031
              3    1961      .      .
              4    1962      .      .
              5    1963      .      .
              6    1964      .      .
              7    1965      .      .
              8    1966      .      .
              9    1967      .      .
             10    1968      .      .
             11    1969      .      .
             12    1970      .      .
             13    1971      .      .
             14    1972      .      .
             15    1973      .      .
             16    1974      .      .
             17    1975      .      .
             18    1976      .      .
             19    1977      .      .
             20    1978      .      .
             21    1979      .      .
             22    1980    100      .
             23    1981    101      .
             24    1982    102      .
             25    1983    103      .
             26    1984    104      .
             27    1985    105      .
             28    1986    106      .
             29    1987    107      .
             30    1988    109      .
             31    1989    111      .
```

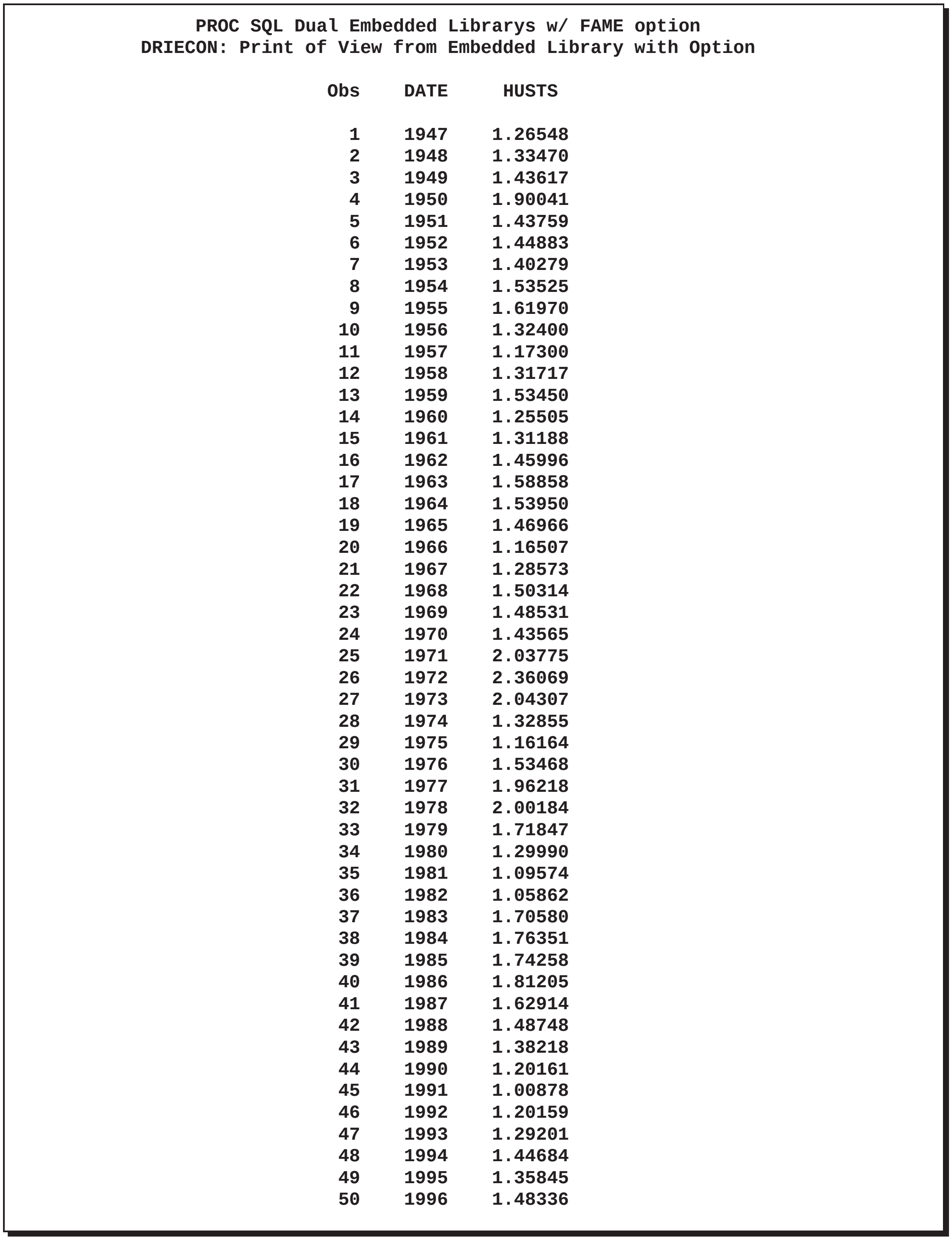
```
/* DRIECON Database */
/*******************/

title2 'DRIECON: Dual Embedded Library Allocations
                 with FAME Option';
proc sql;
   create view fameview as
   select date, husts
   from lib1.driecon
   using libname lib1 sasefame '/economic/databases/testdat'
                      convert=(tech=constant freq=annual),
         libname temp '/usr/local/scratch/mine'
   quit;
```

```
title2 ’DRIECON: Print of View from Embedded Library
                 with FAME Option’;
proc print data=fameview;
run;
```

Note that the SAS option VALIDVARNAME=ANY was used at the top of this example due to special characters being present in the time series names. The output from this example shows how each fameview is the output of the SASEFAME engine’s processing. Note that different engine options could have been used in the USING LIBNAME clause if desired.

Output 5.5.4. Printout of the fameview of DRI Basic Ecomonic data



```
             PROC SQL Dual Embedded Librarys w/ FAME option
        DRIECON: Print of View from Embedded Library with Option

                        Obs    DATE     HUSTS

                          1    1947    1.26548
                          2    1948    1.33470
                          3    1949    1.43617
                          4    1950    1.90041
                          5    1951    1.43759
                          6    1952    1.44883
                          7    1953    1.40279
                          8    1954    1.53525
                          9    1955    1.61970
                         10    1956    1.32400
                         11    1957    1.17300
                         12    1958    1.31717
                         13    1959    1.53450
                         14    1960    1.25505
                         15    1961    1.31188
                         16    1962    1.45996
                         17    1963    1.58858
                         18    1964    1.53950
                         19    1965    1.46966
                         20    1966    1.16507
                         21    1967    1.28573
                         22    1968    1.50314
                         23    1969    1.48531
                         24    1970    1.43565
                         25    1971    2.03775
                         26    1972    2.36069
                         27    1973    2.04307
                         28    1974    1.32855
                         29    1975    1.16164
                         30    1976    1.53468
                         31    1977    1.96218
                         32    1978    2.00184
                         33    1979    1.71847
                         34    1980    1.29990
                         35    1981    1.09574
                         36    1982    1.05862
                         37    1983    1.70580
                         38    1984    1.76351
                         39    1985    1.74258
                         40    1986    1.81205
                         41    1987    1.62914
                         42    1988    1.48748
                         43    1989    1.38218
                         44    1990    1.20161
                         45    1991    1.00878
                         46    1992    1.20159
                         47    1993    1.29201
                         48    1994    1.44684
                         49    1995    1.35845
                         50    1996    1.48336
```

References

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

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

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

FAME Information Services (1995) *Reference Guide to Seamless C HLI*, Ann Arbor, Michigan

FAME Information Services (1995) *Command Reference for Release 7.6, Vols 1 and 2*, Ann Arbor, Michigan

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

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

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

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

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

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

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

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