SAS/ETS® 13.2 User's Guide
The SASEHAVR Interface Engine

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.

Chapter 42
The SASEHAVR Interface Engine

Contents

Overview: SASEHAVR Interface Engine

The SASEHAVR interface engine is a seamless interface between Haver Analytics and SAS data processing that enables SAS users to read economic and financial time series data that reside in a Haver Analytics DLX (Data Link Express) database. The Haver Analytics DLX economic and financial database offerings include U.S. economic indicators, specialized databases, and financial indicators; data about industry, industrial countries, emerging markets, and international organizations; forecasts and as-reported data; and data about U.S. regional services. For more information, see the section "Data Elements Reference: Haver Analytics DLX Database Profile" on page 2974.

The SASEHAVR engine uses the LIBNAME statement to enable you to specify how to subset your Haver data and how to aggregate the selected time series at the same frequency. You can then use the SAS DATA step to perform further subsetting and to store the resulting time series in a SAS data set. You can perform more analysis (if desired) either in the same SAS session or in a later session.

The SASEHAVR engine supports both 32-bit and 64-bit Windows hosts. Haver Analytics supplies two versions of the DLX application programming interface (API), one for 32-bit applications (dlxapi32.dll) and one for 64-bit applications (dlxapi64.dll). Choose the appropriate application, either 32-bit or 64-bit, for your platform. You can follow the instructions for setting up your installation of the Haver API in the section "Setting Up the Haver Analytics DLX Application Programming Interface" on page 2962.

Getting Started: SASEHAVR Interface Engine

Setting Up the Haver Analytics DLX Application Programming Interface

If this is your first time using the SASEHAVR interface engine on your Windows machine, then it is necessary to follow these setup instructions. If you have already used the SASEHAVR interface, then just check the file version number of your already installed dlxapi32.dll (or dlxapi64.dll). For 32-bit installations, the file version is 1.1.9.0, and for 64-bit installations, the file version is 2.0.0.1. The Haver API version number appears in the SAS log the first time you assign a SASEHAVR libref. In Windows Explorer, you can see a file's properties, including its version number, by hovering the mouse pointer over the file icon. Alternatively, you can right-click on the file icon to bring up the properties and click the **Details** tab to see the version number.

To set up the Haver Analytics API on your machine, visit the SAS Technical Support download site at the following URL:

http://ftp.sas.com/techsup/download/base/

First, create a folder on your system drive (usually designated as C:), and name the folder HAVER. Create an environment variable named HAVER as follows:

```
HAVER=C:\HAVER\
```

If your SAS system is 32-bit, then download the files dlxapi32.h, dlxapi32.dll, and dlxapi32.lib to your HAVER folder. If your SAS system is 64-bit, then download the files dlxapi64.h, dlxapi64.dll, and dlxapi64.lib

to your HAVER folder. Second, prepend the location of your HAVER folder to the system environment variable (%PATH%) as follows, so that the SASEHAVR engine can find your downloaded Haver API files:

```
PATH=C:\HAVER\;%PATH%
```

Reboot your system to complete the Haver API setup.

Structure of a SAS Data Set That Contains Time Series Data

SAS represents time series data in 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 time ID variables. The SASEHAVR engine provides a time ID variable called DATE. The DATE variable can be represented in any of the time intervals shown in the section "Mapping Haver Frequencies to SAS Time Intervals" on page 2970.

Reading and Converting Haver DLX Time Series

The SASEHAVR engine supports reading and converting all selected time series that reside in Haver DLX databases. The SASEHAVR engine enables you to limit the range of data by specifying the START= and END= options in the LIBNAME statement. Start dates and end dates are recommended to help save resources when you are processing large databases or a large number of observations.

The SASEHAVR engine enables you to convert or aggregate all selected time series to a desired frequency. By default, the SASEHAVR engine selects the time series variables that match the frequency of the first selected variable. To select variables of one specific frequency, use the FREQ= option. If no selection criteria are specified, the first selected variable is the first physical DLX record read from the Haver database. To force aggregation of all selected variables to the frequency specified by the FREQ= option, use the FORCE=FREQ option. The AGGMODE= option enables you to specify a strict or relaxed aggregation method; by default, AGGMODE=RELAXED. Aggregation is supported only from a more frequent time interval to a less frequent time interval, such as from weekly to monthly. If a conversion to a more frequent frequency is attempted, all missing values are returned by the Haver DLX API. For more information, see the section "Aggregating to Quarterly Frequency Using the FORCE=FREQ Option" on page 2973. The FORCE= option is ignored if the FREQ= option is not specified.

Using the SAS DATA Step

If desired, you can store your selected time series in a SAS data set by using the SAS DATA step. You can further subset your data by using the WHERE, KEEP, or DROP statement in your DATA step.

For more efficient subsetting of time series by Haver variables, Haver groups, Haver sources, Haver short sources, Haver long sources, or Haver geographic codes, you can use the corresponding KEEP=, GROUP=, SOURCE=, SHORTSOURCE=, LONGSOURCE=, GEOGCODE1=, or GEOGCODE2= option in the LIBNAME *libref* SASEHAVR statement. To see the available Haver selection key values, including geographic codes, short sources, and long sources for your database, specify the OUTSELECT=ON option.

From the OUTSELECT= option output, you can use convenient wildcard symbols to create the selection list for your next LIBNAME *libref* SASEHAVR statement.

There are three wildcard symbols: '*', '?' and '#'. The '*' wildcard corresponds to any character string and includes any string pattern that corresponds to that position in the matching variable name. The '?' stands for any single alphanumeric character. Lastly, the '#' wildcard corresponds to a single numeric character.

You can also deselect time series by Haver variables, by Haver groups, by Haver sources, by Haver short sources, by Haver long sources, or by Haver geographic codes, by using the corresponding DROP=, DROPGROUP=, DROPSOURCE=, DROPSHORT=, DROPLONG=, DROPGEOG1=, or DROPGEOG2= option. These options also support wildcards.

After your selected data is stored in a SAS data set, you can use it as you would any other SAS data set.

Using the SAS Windowing Environment

You can see the available data sets in the SAS LIBNAME window of the SAS windowing environment by selecting the SASEHAVR *libref* in the LIBNAME window that you have previously defined in your LIBNAME statement. You can view your SAS output observations by double-clicking on the desired output data set *libref* in the LIBNAME window of the SAS windowing environment. You can type **Viewtable** on the SAS command line to view your SASEHAVR tables, views, or librefs.

Before you use **Viewtable**, it is recommended that you store your output data sets in a physical folder or library that is separate from the folder or library used for your input databases. (The default location for output data sets is the SAS Work library.) If you do not follow this guideline, you will receive the following error message for each input database that does not have the selected options in the SASEHAVR *libref* that you double-clicked:

```
ERROR: No variable selected with current options.
```

Syntax: SASEHAVR Interface Engine

The SASEHAVR engine uses standard engine syntax. Table 42.1 summarizes the options used in the LIBNAME *libref* SASEHAVR statement.

Table 42.1 Summary of LIBNAME *libref* SASEHAVR Statement Options

Option	Description
FREQUENCY=	Specifies the Haver frequency
START=	Specifies a Haver start date to limit the selection of time series to those that begin with the specified date
END=	Specifies a Haver end date to limit the selection of time series to those that end with the specified date
KEEP=	Specifies a list of comma-delimited Haver variables to keep in the output SAS data set

Table 42.1 *continued*

Option	Description
DROP=	Specifies a list of comma-delimited Haver variables to drop from the output SAS data set
GROUP=	Specifies a list of comma-delimited Haver groups to keep in the output SAS data set
DROPGROUP=	Specifies a list of comma-delimited Haver groups to drop from the output SAS data set
SOURCE=	Specifies a list of comma-delimited Haver sources to keep in the output SAS data set
DROPSOURCE=	Specifies a list of comma-delimited Haver sources to drop from the output SAS data set
SHORT=	Specifies a list of comma-delimited Haver short sources to keep in the output SAS data set
DROPSHORT=	Specifies a list of comma-delimited Haver short sources to drop from the output SAS data set
LONG=	Specifies a list of comma-delimited Haver long sources to keep in the output SAS data set
DROPLONG=	Specifies a list of comma-delimited Haver long sources to drop from the output SAS data set
GEOG1=	Specifies a list of comma-delimited Haver geography1 codes to keep in the output SAS data set
DROPGEOG1=	Specifies a list of comma-delimited Haver geography1 codes to drop from the output SAS data set
GEOG2=	Specifies a list of comma-delimited Haver geography2 codes to keep in the output SAS data set
DROPGEOG2=	Specifies a list of comma-delimited Haver geography2 codes to drop from the output SAS data set
OUTSELECT=	Specifies what values the output data is to contain
FORCE=FREQ	Specifies that all selected time series variables be aggregated to the frequency specified in the FREQ= option
AGGMODE=	Specifies the aggregation method used for aggregating time series (STRICT or RELAXED)

LIBNAME *libref* SASEHAVR Statement

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

The '*physical name*' specifies the location of the folder where your Haver DLX database resides.

You can use the following options in the LIBNAME *libref* SASEHAVR statement:

FREQ=*haver_frequency*

FREQUENCY=*haver_frequency*

INTERVAL=*haver_frequency*

specifies the Haver frequency. All Haver frequencies are supported by the SASEHAVR engine. Accepted frequency values are annual, year, yearly, quarter, quarterly, qtr, monthly, month, mon, week.1, week.2, week.3, week.4, week.5, week.6, week.7, weekly, week, daily, and day.

START=*start_date*

STARTDATE=*start_date*

STDATE=*start_date*

BEGIN=*start_date*

specifies the start date for the time series in the form YYYYMMDD.

END=*end_date*

ENDDATE=*end_date*

ENDATE=*end_date*

specifies the end date for the time series in the form YYYYMMDD.

KEEP="*haver_variable_list*"

specifies the list of Haver variables to be included in the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

DROP="*haver_variable_list*"

specifies the list of Haver variables to be excluded from the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

GROUP="*haver_group_list*"

KEEPGROUP="*haver_group_list*"

specifies the list of Haver groups to be included in the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

DROPGROUP="*haver_group_list*"

specifies the list of Haver groups to be excluded from the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

SOURCE="*haver_source_list*"

KEEPSOURCE="*haver_source_list*"

specifies the list of Haver sources to be included in the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

DROPSOURCE="*haver_source_list*"

specifies the list of Haver sources to be excluded from the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

SHORT="*haver_shortsource_list*"

KEEPSHORT="*haver_shortsource_list*"

SHORTSOURCE="*haver_shortsource_list*"

specifies the list of Haver short sources to be included in the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

DROPSHORT="*haver_shortsource_list*"

DROPSHORTSOURCE="*haver_shortsource_list*"

specifies the list of Haver short sources to be excluded from the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

LONG="*haver_longsource_list*"

KEEPLONG="*haver_longsource_list*"

LONGSOURCE="*haver_longsource_list*"

specifies the list of Haver long sources to be included in the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

DROPLONG="*haver_longsource_list*"

DROPLONGSOURCE="*haver_longsource_list*"

specifies the list of Haver long sources to be excluded from the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

GEOG1="*haver_geographycode1_list*"

KEEPGEOG1="*haver_geographycode1_list*"

GEOGCODE1="*haver_geographycode1_list*"

specifies the list of Haver geography1 codes to be included in the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

DROPGEOG1="*haver_geographycode1_list*"

DROPGEOGCODE1="*haver_geographycode1_list*"

specifies the list of Haver geography1 codes to be excluded from the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

GEOG2="*haver_geographycode2_list*"

KEEPGEOG2="*haver_geographycode2_list*"

GEOGCODE2="*haver_geographycode2_list*"

specifies the list of Haver geography2 codes to be included in the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

DROPGEOG2="*haver_geographycode2_list*"

DROPGEOGCODE2="*haver_geographycode2_list*"

specifies the list of Haver geography2 codes to be excluded from the output SAS data set. This list is comma-delimited and must be surrounded by quotes "".

OUTSELECT=ON | OFF

specifies what the output data set shows. OUTSELECT=ON specifies that the output data set show values of selection keys (such as geography codes, groups, sources, short sources, and long sources) for each selected variable name (time series) in the database. OUTSELECT=OFF specifies that the output data set show the observations in the range for all selected time series. The default is OUTSELECT=OFF.

AGGMODE=STRICT | RELAXED

specifies whether the SASEHAVR engine uses a strict or relaxed aggregation method when converting time series from a higher to lower frequency.

A strict aggregation method returns a missing value whenever there is a missing observation in a time period. For instance, if a monthly time series has a missing value for the month of February, 2005, then attempting to aggregate to a quarterly frequency results in a missing value for the first quarter of 2005. The SAS log reports the status of this option.

When a relaxed aggregation method is used, some observations can be missing, but the relaxed method returns an aggregated value calculated from the nonmissing data points according to the series aggregation type (average, sum, or end of period). Average type only needs one valid (nonmissing) data point to calculate the average. Sum type needs all the data points to be available in order to sum the values. End of period type calculates the end of period value if there is at least one valid (nonmissing) data point in the aggregated span. It returns the last available valid data point in the aggregated span. The default is AGGMODE=RELAXED.

FORCE=FREQ

specifies that the selected variables be aggregated to the frequency in the FREQ= option. Aggregation is supported only from a more frequent time interval to a less frequent time interval, such as from weekly to monthly. See the section "Aggregating to Quarterly Frequency Using the FORCE=FREQ Option" on page 2973 for sample output and suggested error recovery from attempting a conversion that yields missing values when a higher frequency conversion is specified. This option is ignored if the FREQ= option is not set. For a more complete discussion of Haver frequencies and SAS time intervals, see the section "Mapping Haver Frequencies to SAS Time Intervals" on page 2970.

Following is an example of the LIBNAME *libref* SASEHAVR statement:

```
LIBNAME libref sasehavr 'physical-name'
        FREQ=MONTHLY;
```

By default, the SASEHAVR engine reads all time series in the Haver database that you reference by *libref*. The *start_date* is specified in the form YYYYMMDD. The start date is used to delimit the data to a specified start date.

For example, to read the time series in the TEST library starting on July 4, 1996, specify the following statement:

```
LIBNAME test sasehavr 'physical-name'
        STARTDATE=19960704;
```

When you use the START= option, you limit the range of observations that are read from the time series and that are converted to the desired frequency. Start dates can help save resources when processing large databases or when processing a large number of observations. It is also possible to select specific variables to be included or excluded from the SAS data set by using the KEEP= or the DROP= option, respectively.

```
LIBNAME test sasehavr 'physical-name'
        KEEP="ABC*, XYZ??";

LIBNAME test sasehavr 'physical-name'
        DROP="*SC*, #T#";
```

When the KEEP= or the DROP= option is used, the resulting SAS data set keeps or drops the variables that you select in that option. Three wildcards are available: '*', '?' and '#'. The '*' wildcard corresponds to any character string and includes any string pattern that corresponds to that position in the matching variable name. The '?' means that any single alphanumeric character is valid. The '#' wildcard corresponds to a single numeric character. You can also select time series in your data by using the GROUP=, SOURCE=, SHORT=, LONG=, GEOG1=, or the GEOG2= option to select on group name, source name, short source name, long source name, geography1 code, or the geography2 code, respectively. Alternatively, you can deselect time series by using the DROPGROUP=, DROPSOURCE=, DROPSHORT=, DROPLONG=, DROPGEOG1=, or the DROPGEOG2= option, respectively.

Following are examples that perform variable selection (or deselection) based on groups or sources:

```
LIBNAME test sasehavr 'physical-name'
        GROUP="CBA, *ZYX";

LIBNAME test sasehavr 'physical-name'
        DROPGROUP="TKN*, XCZ?";

LIBNAME test sasehavr 'physical-name'
        SOURCE="FRB";

LIBNAME test sasehavr 'physical-name'
        DROPSOURCE="NYSE";
```

SASEHAVR selects only the variables that are of the specified frequency in the FREQ= option. If this option is not specified, SASEHAVR selects the variables that match the frequency of the first selected variable. If no other selection criteria are specified, by default the first selected variable is the first physical DLX record read from the Haver database. You can specify the FORCE=FREQ option to force the aggregation of all variables selected to be of the frequency specified in the FREQ= option. Aggregation is supported only from a more frequent time interval to a less frequent time interval, such as from weekly to monthly. See the section "Aggregating to Quarterly Frequency Using the FORCE=FREQ Option" on page 2973 for suggested recovery from using a frequency that does not aggregate the data appropriately. The FORCE= option is ignored if the FREQ= option is not specified. The AGGMODE= STRICT option is used when a strict aggregation method is desired. The default value for AGGMODE is RELAXED, the same method that was used in prior releases of SASEHAVR.

Details: SASEHAVR Interface Engine

SAS Output Data Set

You can use the SAS DATA step to write the Haver converted series to a SAS data set so that you can easily analyze the data using the SAS System. You can specify the name of the output data set in the DATA statement. This causes the engine supervisor to create a SAS data set with the specified name in either the SAS Work library, or if specified, the Sasuser library.

When OUTSELECT=OFF (the default), the contents of the SAS data set include the date of each observation, the name of each series read from the Haver database, and the label or Haver description of each series. Missing values are represented as '.' in the SAS data set. You can use the PRINT procedure and the CONTENTS procedure to print your output data set and its contents. You can use the SQL procedure along with the SASEHAVR engine to create a view of your SAS data set.

The DATE variable in the SAS data set contains the date of the observation. The SASEHAVR engine automatically maps the Haver intervals to the appropriate corresponding SAS intervals.

When OUTSELECT=ON, the OUT= data set does not contain the observations of all time series. Instead, each observation contains the name of the time series, the source of the time series, the geography1 code, the geography2 code, the short source, and the long source for that time series. In addition, the contents of the OUT= data set shows every selected time series name and label. See Output 42.11.1 and Output 42.11.2 for more details about the OUTSELECT=ON option.

A more detailed discussion of how to map Haver frequencies to SAS time intervals follows.

Mapping Haver Frequencies to SAS Time Intervals

Table 42.2 summarizes the mapping of Haver frequencies to SAS time intervals. For more information, see Chapter 4, "Date Intervals, Formats, and Functions."

Table 42.2 Mapping Haver Frequencies to SAS Time Intervals

Haver Frequency	SAS Time Interval	FREQ=
ANNUAL	YEAR	YEARLY
QUARTERLY	QTR	QTRLY
MONTHLY	MONTH	MON
WEEKLY (SUNDAY)	WEEK.1	WEEK.1
WEEKLY (MONDAY)	WEEK.2	WEEK.2
WEEKLY (TUESDAY)	WEEK.3	WEEK.3
WEEKLY (WEDNESDAY)	WEEK.4	WEEK.4
WEEKLY (THURSDAY)	WEEK.5	WEEK.5
WEEKLY (FRIDAY)	WEEK.6	WEEK.6
WEEKLY (SATURDAY)	WEEK.7	WEEK.7
WEEKLY WEEK.1-WEEK.7	WEEKLY	WEEKLY
DAILY	WEEKDAY17W	DAY

Error Recovery for the SASEHAVR Interface Engine

Common errors are easy to avoid by noting the valid dates that are specified in the warning messages in your SAS log. Often you can get rid of errors by removing the date restriction (START= and END= options), by removing the FORCE=FREQ option, or by deleting the FREQ= option so that the frequency defaults to the original frequency rather than attempting a conversion.

Following are some common error scenarios and how to handle them.

Using the Optimum Range for Best Output Results

Suppose you see the following warnings in your SAS log:

```
libname kgs2 sasehavr "%sysget(HAVER_DATA)"
        start= 19550101 end=19600105
        keep="FCSEED, FCSEEI, FCSEEM, BGSX, BGSM, FXDUSBC"
        group="I01, F56, M02, R30"
        source="JPM,CEN,OMB" ;

NOTE: Libref KGS2 was successfully assigned as follows:
      Engine:        SASEHAVR
      Physical Name: C:\haver

data kgse9;
   set kgs2.haver;
NOTE: Defaulting to MONTHLY frequency.
WARNING: Start date (19550101) is not a valid date.
         Engine is ignoring your start date and using
         default. Setting the default Haver start date to 7001.
WARNING: End date (19600105) is not a valid date.
         Engine is ignoring your end date and using
         default. Setting the default Haver end date to 10103.
run;

NOTE: There were 375 observations read from the data set KGS2.HAVER.
NOTE: The data set WORK.KGSE9 has 375 observations and 4 variables.
```

The important diagnostic to note here is the warning message that tells you that the data starts in January of 1970 (Haver date 7001), and ends in March, 2001 (Haver date 10103). Since the specified range falls outside the range of data, no observations are in range. So, the engine uses the default range stated in the warning messages. Change the START= and END= options to overlap the results in data that span from JAN1970 to MAR2001. To view the entire range of selected data, remove the START= and END= options from the LIBNAME statement:

```
libname kgs sasehavr "%sysget(HAVER_DATA)"
        keep="FCSEED, FCSEEI, FCSEEM, BGSX, BGSM, FXDUSBC"
        group="I01, F56, M02, R30"
        source="JPM,CEN,OMB" ;

NOTE: Libref KGS was successfully assigned as follows:
      Engine:        SASEHAVR
      Physical Name: C:\haver

data kgse5;
   set kgs.haver;
NOTE: Defaulting to MONTHLY frequency.
   run;

NOTE: There were 375 observations read from the data set KGS.HAVER.
NOTE: The data set WORK.KGSE5 has 375 observations and 4 variables.
```

Using a Valid Range of Data with START= and END= Options

In this example, an error about an invalid range is issued:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"  freq=Weekly
   start=20060301 end=20060531;
NOTE: Libref LIB1 was successfully assigned as follows:
      Engine:        SASEHAVR
      Physical Name: C:\haver
libname lib2 "\\dntsrc\usrtmp\saskff" ;
NOTE: Libref LIB2 was successfully assigned as follows:
      Engine:        V9
      Physical Name: \\dntsrc\usrtmp\saskff
data lib2.wweek;
   set lib1.intwkly;
ERROR: No observations found inside RANGE.
       The valid range for HAVER dates is (610104-1050318).
ERROR: No observations found in specified range.
            keep date m11: ;
         run;

WARNING: The variable date in the DROP, KEEP, or RENAME list
         has never been referenced.
WARNING: The variable m11: in the DROP, KEEP, or RENAME list
         has never been referenced.
NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set LIB2.WWEEK may be incomplete.
         When this step was stopped there were 0
         observations and 0 variables.
WARNING: Data set LIB2.WWEEK was not replaced because this step was stopped.
```

The important diagnostic message is the first error statement which tells you that the range of Haver dates is not valid for the specified frequency. A valid range is one that overlaps the dates (610104–1050318). Removing the range altogether causes the engine to output the entire range of data.

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"  freq=Weekly;

NOTE: Libref LIB1 was successfully assigned as follows:
      Engine:        SASEHAVR
      Physical Name: C:\haver

libname lib2 "\\dntsrc\usrtmp\saskff" ;
NOTE: Libref LIB2 was successfully assigned as follows:
      Engine:        V9
      Physical Name: \\dntsrc\usrtmp\saskff

data lib2.wweek;
   set lib1.intwkly;
   keep date m11: ;
run;

NOTE: There were 2307 observations read from the data set LIB1.INTWKLY.
NOTE: The data set LIB2.WWEEK has 2307 observations and 35 variables.
```

Since the START= and END= options give day-based dates, it is important to use dates that correspond to the FREQ= option when giving a range of dates, especially with weekly frequencies such as WEEK.1–WEEK.7. Since FREQ=WEEK.4 selects weeks that begin on Wednesday, the start and end dates need to be specified as Wednesday dates.

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"  freq=Week.4
   start=20050302 end=20050309;
NOTE: Libref LIB1 was successfully assigned as follows:
      Engine:        SASEHAVR
      Physical Name: \\tappan\crsp1\haver
title2 'Weekly dataset with freq=week.4 range is small';
libname lib2 "\\dntsrc\usrtmp\saskff" ;
NOTE: Libref LIB2 was successfully assigned as follows:
      Engine:        V9
      Physical Name: \\dntsrc\usrtmp\saskff

data lib2.wweek;
   set lib1.intwkly;
   keep date m11: ;
run;

NOTE: There were 2 observations read from the data set LIB1.INTWKLY.
NOTE: The data set LIB2.WWEEK has 2 observations and 25 variables.
```

Giving bad dates (for example, Tuesday dates) for a Wednesday FREQ=WEEK.4 results in the following error:

```
ERROR: Fatal error in GetDate routine.
       Remove the range statement or change the START= date to
       be consistent with the freq=option.
ERROR: No observations found in specified range.
```

Aggregating to Quarterly Frequency Using the FORCE=FREQ Option

In the next example, six time series are selected by the KEEP= option. Their frequencies are annual, monthly, and quarterly, so when the FREQ=WEEKLY and FORCE=FREQ options are used, a diagnostic appears in the log stating that the engine is forcing the frequency to QUARTERLY for better date alignment of observations. The first selected variable is BALO, which is a quarterly time series and causes the default choice of FREQ to be quarterly.

```
title1 '***HAVKWC.SAS: KEEP= option tests with wildcards***';

%setup( ets );

 /*----------------*/
 /* Wildcard: *    */
 /*----------------*/

title2 "keep=B*, G*, I*";
title3 "6 valid variables are: BALO BGSM BGSX BPBCA G IUM";
```

```
libname lib1 sasehavr 'C:\haver\' keep="B*, G*, I*"
   freq=weekly force=freq;
NOTE: Libref LIB1 was successfully assigned as follows:
      Engine:        SASEHAVR
      Physical Name: C:\haver\

data wc;
   set lib1.haver;
WARNING: Earliest Start Date in DLX Database matches QUARTERLY frequency
         better than the specified WEEKLY frequency.
         Engine is forcing the frequency to QUARTERLY for better date
         alignment of observations.
run;

NOTE: There were 221 observations read from the data set LIB1.HAVER.
NOTE: The data set WORK.WC has 221 observations and 7 variables.
```

Note that the time series IUM is an annual frequency. The attempt to convert to a quarterly frequency produces all missing values in the output range because aggregation produces only missing values when forced to go from a lower frequency to a higher frequency.

Data Elements Reference: Haver Analytics DLX Database Profile

The Haver DLX economic and financial database offerings include U.S. economic indicators, specialized databases, financial indicators, industry, industrial countries, emerging markets, international organizations, forecasts and as-reported data, and U.S. regional service. Table 42.3 is a list of available databases and the corresponding description of each.

Table 42.3 Available Data Offerings

Database Name	Offering Type	Description
USECON	U.S. economic indicators	U.S. economic, financial data
USNA	U.S. economic indicators	Complete U.S. NIPA accounts from the Bureau of Economic Analysis (BEA)
SURVEYS	U.S. economic indicators	Business and consumer expectations, surveys
SURVEYW	U.S. economic indicators	Business and consumer expectations, weekly surveys
CPIDATA	U.S. economic indicators	Consumer price indexes (CPI), monthly in CPI detailed report
PPI	U.S. economic indicators	Producer price indexes (PPI), by the Bureau of Labor Statistics (BLS)
PPIR	U.S. economic indicators	Producer price indexes by BLS
LABOR	U.S. economic indicators	Employment and earnings by BLS

Table 42.3 *continued*

Database Name	Offering Type	Description
EMPL	U.S. economic indicators	Household employment survey, monthly by BLS
CEW	U.S. economic indicators	Covered employment and wages, monthly, quarterly
IP	U.S. economic indicators	Industrial production and capacity utilization by Federal Reserve Board (FRB)
FFUNDS	U.S. economic indicators	Flow of funds data by FRB
CAPSTOCK	U.S. economic indicators	Capital stock by the Bureau of Economic Analysis (BEA)
USINT	U.S. economic indicators	U.S. international trade (TIC) data by country and product
CBDB	Specialized databases	Conference Board database, monthly by The Conference Board (TCB)
BCI	Specialized databases	U.S. business cycle indicators, by TCB
UMSCA	Specialized databases	Consumer Sentiment Survey from the University of Michigan
FIBERUS	Specialized databases	U.S. FIBER business cycle indicators from the Foundation of International Business and Economic Research (FIBER)
FIBER	Specialized databases	FIBER business cycle indicators from FIBER
DAILY	Financial indicators	U.S. daily statistics data
INTDAILY	Financial indicators	Country daily statistics
WEEKLY	Financial indicators	U.S. weekly statistics
INTWKLY	Financial indicators	Country weekly statistics
SPD	Financial indicators	Standard and Poor's industry groups, daily
SPW	Financial indicators	Standard and Poor's industry groups, weekly
SPM	Financial indicators	Standard and Poor's industry groups, monthly
SPAH	Financial indicators	Standard and Poor's Analysts' Handbook, yearly
MSCID	Financial indicators	Morgan Stanley Capital International, daily
MSCIW	Financial indicators	Morgan Stanley Capital International, weekly

Table 42.3 *continued*

Database Name	Offering Type	Description
MSCIM	Financial indicators	Morgan Stanley Capital International, monthly
EMBI	Financial indicators	Emerging Markets Bond Index from J.P. Morgan
BONDINDX	Financial indicators	U.S. bond indexes, from Barclays Capital, Citigroup, Merrill Lynch, and Standard and Poors
BONDS	Financial indicators	Citigroup bond performance indexes by Citigroup Global Markets, formerly Salomon Smith Barney
ICI	Financial indicators	Mutual fund activity from the Investment Company Institute
QFR	Financial indicators	Quarterly financial report by FRB
MBAMTG	Financial indicators	Mortgage delinquency rates by the Mortgage Bankers Association
MBAMOS	Financial indicators	Mortgage origination surveys, from two Mortgage Banker Association surveys
MARKIT	Financial indicators	Markit's indexes used to price credit default swaps
DLINQ	Financial indicators	Consumer delinquency rates by American Bankers Association, monthly
FDIC	Financial indicators	FDIC banking statistics TIC data from the Quarterly Banking Profile
GOVFIN	Financial indicators	U.S. government financial statistics by U.S. Treasury
INDUSTRY	Industry	U.S. industry statistics, from Department of Agriculture, trade associations
WARDS	Industry	Automotive statistics, from Ward's Automotive Group
USDA	Industry	World agriculture statistics, from U.S. Department of Agriculture (USDA)
REALTOR	Industry	Home sales from National Association of Realtors
CREALTOR	Industry	Home sales from National Association of Realtors
PREALTOR	Industry	Pending home sales from National Association of Realtors
EEI	Industry	U.S. electric output from the Edison Electric Institute, weekly
ASM	Industry	Annual Survey of Manufactures from the U.S. Census Bureau
RAILSHAR	Industry	Railcar loadings from Association of American Railroads and Atlantic Systems
CHEMWEEK	Industry	Weekly chemical prices from Access Intelligence
BALTIC	Industry	Baltic freight indexes, from the Baltic Exchange in London

Table 42.3 *continued*

Database Name	Offering Type	Description
OGJ	Industry	U.S. and international energy statistics, from Pennwell Publishing's Oil & Gas Journal
OGJANN	Industry	U.S. and international energy statistics, from Pennwell Publishing's Oil & Gas Journal, annual
OILWKLY	Industry	Weekly oil statistics, from Penwell Publishing's Oil & Gas Journal
OMI	Industry	Oil market intelligence, from Energy Intelligence
NGW	Industry	Natural Gas Week, from Energy Intelligence
WGI	Industry	World Gas Intelligence, from Energy Intelligence
G10+	Industrial countries	International Macroeconomic Data by Haver Analytics
PMI	Industrial countries	Purchasing Managers Indexes by Markit Economics
INTSRVYS	Industrial countries	Country surveys
JAPAN	Industrial countries	Japan from Nomura Research Institute
JAPANW	Industrial countries	Japan from Nomura Research Institute, weekly
CANSIM	Industrial countries	Canada from Statistics Canada and the Bank of Canada
CANSIMR	Industrial countries	Canada from Statistics Canada and the Bank of Canada
UK	Industrial countries	United Kingdom, from the Office of National Statistics and the Bank of England
UKSRVYS	Industrial countries	United Kingdom surveys, by NTC Economics, Ltd.
GERMANY	Industrial countries	Germany, from the Deutsche Bundesbank and Statistics Bundesamt
FRANCE	Industrial countries	France, Statistics from INSEE (France's National Statistical Office), the Bank of France, and the Ministry of France
ITALY	Industrial countries	Italy, from Istituto Nazionale di Statistica and Banca d'Italia
SPAIN	Industrial countries	Spain, from the Instituto Nacional de Estadistica and the Banco de Espana
IRELAND	Industrial countries	Ireland, from the Central Statistics Office and Central Bank
NORDIC	Industrial countries	Norway, Sweden, Denmark, Finland
ALPMED	Industrial countries	Austria, Switzerland, Greece, Portugal

Table 42.3 *continued*

Database Name	Offering Type	Description
BENELUX	Industrial countries	Belgium, Netherlands, Luxembourg, monthly
ANZ	Industrial countries	Australia and New Zealand
EMERGELA	Emerging markets	Latin American macroeconomic data
EMERGECW	Emerging markets	Central and Eastern Europe and Western Asia
EMERGEMA	Emerging markets	Middle East and African emerging markets
EMERGEPR	Emerging markets	Asia/Pacific Rim emerging markets
CHINA	Emerging markets	CEIC Premium China Database, from CEIC Data Company Ltd (CEIC)
INDIA	Emerging markets	CEIC Premium India Database, from CEIC
EUROSTAT	International organizations	European Union data from Eurostat, the European Central Bank, and the European Commission
EULABOR	International organizations	European Union regional labor from Eurostat
OECDMEI	International organizations	Organisation for Economic Cooperation and Development (OECD) main economic indicators
OECDNAQ	International organizations	OECD Quarterly National Accounts
OECDNA	International organizations	OECD Annual National Accounts
OECDFIN	International organizations	OECD Financial Accounts and Financial Balance Sheets
OECDFEI	International organizations	OECD foreign direct investment data
OUTLOOK	International organizations	OECD Economic Outlook
IFS	International organizations	International Financial Statistics from International Monetary Fund
IFSANN	International organizations	International Financial Statistics, annual from International Monetary Fund
IMFBOP	International organizations	Balance of Payment Statistics from International Monetary Fund
IMFBOPA	International organizations	Annual Balance of Payment Statistics from International Monetary Fund
IMFDOT	International organizations	Direction of Trade Statistics from International Monetary Fund

Table 42.3 *continued*

Database Name	Offering Type	Description
IMFDOTM	International organizations	Direction of Trade Statistics, monthly from International Monetary Fund
IMFWEO	International organizations	Analysis and projections of economic development at global level from International Monetary Fund
BIS	International organizations	International financial claims and liabilities from the Bank for International Settlements
WBPRICES	International organizations	World commodity prices from The World Development Prospects Group (Pinksheets)
WBDEBT	International organizations	Global development finance from The World Bank debt tables
UNPOP	International organizations	United Nations population projections
INTCOMP	International organizations	International comparisons from U.S. Bureau of Labor Statistics
MA4CAST	Forecasts and as-reported data	Short-term U.S. economic forecasts from Macroeconomic Advisers
MA4CSTL	Forecasts and as-reported data	Long-term U.S. economic forecasts from Macroeconomic Advisers
CQM	Forecasts and as-reported data	Canadian quarterly model from Centre for Spatial Economics
CPM	Forecasts and as-reported data	Canadian provincial model from Centre for Spatial Economics
OEFQMACR	Forecasts and as-reported data	Global macroeconomic forecasts from Oxford Economic Forecasting
OEFMAJOR	Forecasts and as-reported data	Global macroeconomic forecasts from Oxford Economic Forecasting
OEFINTER	Forecasts and as-reported data	Global macroeconomic forecasts from Oxford Economic Forecasting
OEFMINOR	Forecasts and as-reported data	Global macroeconomic forecasts from Oxford Economic Forecasting
OEFQIND	Forecasts and as-reported data	Global industry from Oxford Economic Forecasting
EIUIAMER	Forecasts and as-reported data	Market indicators and forecasts (America) from Economist Intelligence Unit
EIUIASIA	Forecasts and as-reported data	Market indicators and forecasts (Asia) from Economist Intelligence Unit
EIUIEEUR	Forecasts and as-reported data	Market indicators and forecasts (Eastern Europe) from Economist Intelligence Unit
EIUIMENA	Forecasts and as-reported data	Market indicators and forecasts from Economist Intelligence Unit
EIUISUBS	Forecasts and as-reported data	Market indicators and forecasts from Economist Intelligence Unit

Table 42.3 *continued*

Database Name	Offering Type	Description
EIUIWEUR	Forecasts and as-reported data	Market indicators and forecasts (Western Europe) from Economist Intelligence Unit
EIUIREGS	Forecasts and as-reported data	Market indicators and forecasts from Economist Intelligence Unit
EIUDAMER	Forecasts and as-reported data	Country data (America) from Economist Intelligence Unit
EIUDASIA	Forecasts and as-reported data	Country data (Asia) from Economist Intelligence Unit
EIUDEEUR	Forecasts and as-reported data	Country data (Eastern Europe) from Economist Intelligence Unit
EIUDMENA	Forecasts and as-reported data	Country data from Economist Intelligence Unit
EIUDSUBS	Forecasts and as-reported data	Country data from Economist Intelligence Unit
EIUDWEUR	Forecasts and as-reported data	Country data (Western Europe) from Economist Intelligence Unit
EIUDOECD	Forecasts and as-reported data	Country data (OECD) from Economist Intelligence Unit
EIUDREGS	Forecasts and as-reported data	Country data from Economist Intelligence Unit
AS1REPNA	Forecasts and as-reported data	Action Economics forecast medians and as reported data
MMSAMER	Forecasts and as-reported data	MMS survey medians and as-first-reported data (America) from MMS International
MMSEUR	Forecasts and as-reported data	MMS survey medians and as-first-reported data (Europe) from MMS International
SURVEYS	Forecasts and as-reported data	Economic survey forecasts
AS4CAST	Forecasts and as-reported data	Historical economic forecasts
ASREPGDP	Forecasts and as-reported data	As-reported U.S. gross domestic product from Bureau of Economic Analysis
LABORR	U.S. regional	Monthly payroll employment from Bureau of Labor Statistics
EMPLR	U.S. regional	Labor force and unemployment from Bureau of Labor Statistics
EMPLC	U.S. regional	Labor force and unemployment from Bureau of Labor Statistics
BEAEMPL	U.S. regional	Annual employment by industry
BEAEMPM	U.S. regional	Annual employment by industry
PERMITS	U.S. regional	Residential building permits
PERMITY	U.S. regional	Residential building permits
PERMITP	U.S. regional	Residential building permits

Table 42.3 *continued*

Database Name	Offering Type	Description
PERMITC	U.S. regional	Residential building permits
PERMITA	U.S. regional	Residential building permits
REGIONAL	U.S. regional	Selected regional indicators
REGIONW	U.S. regional	Selected regional indicators
PIQR	U.S. regional	Personal income
PIR	U.S. regional	Personal income
PIRMSA	U.S. regional	Personal income
PICOUNTY	U.S. regional	Personal income
PIRC1 to 9	U.S. regional	Personal income
MBAMTG	U.S. regional	Mortgage delinquency rates from Mortgage Bankers Association
DLINQR	U.S. regional	Consumer delinquency rates from American Bankers Association
FALOAN	U.S. regional	Real estate and construction delinquency rates by Foresight Analytics
BANKRUPT	U.S. regional	Bankruptcies by county and metropolitan statistical area
GSP	U.S. regional	Gross state product from BEA
GDPMSA	U.S. regional	Gross domestic product by Metropolitan Statistical Areas (MSA)
USPOP	U.S. regional	Population by age and sex
USPOPC	U.S. regional	Population by age and sex
PORTS	U.S. regional	Trade by port
EXPRQ1 to 9	U.S. regional	Exports by industry and country from the World Institute for Strategic Economic Research and the U.S. Census Bureau
EXPORTSR	U.S. regional	Exports by industry and country from the World Institute for Strategic Economic Research and the U.S. Census Bureau
GOVFINR	U.S. regional	Government financial statistics from the U.S. Census Bureau and Rockefeller Institute of Government
FDICR	U.S. regional	FDIC banking statistics

Examples: SASEHAVR Interface Engine

Before running the following sample code, set your HAVER_DATA environment variable to point to the SAS/ETS SASMISC folder that contains sample Haver databases. The provided sample data files are HAVERD.DAT, HAVERD.IDX, HAVERW.IDX, and HAVERW.DAT. In the following example, the Haver database is called haverw and it resides in the directory lib1. The DATA statement names the SAS output data set hwouty, which will reside in the Work library.

Example 42.1: Examining the Contents of a Haver Database

To see which time series are in your Haver database, use the CONTENTS procedure with the SASEHAVR LIBNAME statement to read the contents.

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=yearly start=19920101
        end=20041231
        force=freq;

data hwouty;
   set lib1.haverw;
run;

title1 'Haver Analytics Database, HAVERW.DAT';
title2 'PROC CONTENTS for Time Series converted to yearly frequency';
proc contents data=hwouty;
run;
```

All time series in the Haver haverw database are listed alphabetically in Output 42.1.1.

Output 42.1.1 Examining the Contents of Haver Analytics Database, haverw.dat

Haver Analytics Database, HAVERW.DAT
PROC CONTENTS for Time Series converted to yearly frequency

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	DATE	Num	8	YEAR4.	Date of Observation
2	FA	Num	8		Total Assets: All Commercial Banks (SA, Bil.$)
3	FCM1M	Num	8		1-Month Treasury Bill Market Bid Yield at Constant Maturity (%)
4	FM1	Num	8		Money Stock: M1 (SA, Bil.$)
5	FTA1MA	Num	8		Treasury 4-Week Bill: Total Amount Accepted (Bil$)
6	FTB3	Num	8		3-Month Treasury Bills, Auction (% p.a.)
7	LICN	Num	8		Unemployment Insurance: Initial Claims, State Programs (NSA, Thous)

You could also use the following SAS statements to create a SAS data set named hwouty and to print its contents.

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
       freq=yearly
       start=19920101
       end=20041231
       force=freq;

data hwouty;
   set lib1.haverw;
run;

title1 'Haver Analytics Database, Frequency=yearly, infile=haverw.dat';
title2 'Define a range inside the data range for OUT= dataset,';
title3 'Using the START=19920101 END=20041231 LIBNAME options.';

proc print data=hwouty;
run;
```

The preceding LIBNAME LIB1 statement specifies that all time series in the haverw database be converted to a yearly frequency but to select only the range of data from January 1, 1992, to December 31, 2004. The resulting SAS data set hwouty is shown in Output 42.1.2.

Output 42.1.2 Defining a Range inside the Data Range for Yearly Time Series

Haver Analytics Database, Frequency=yearly, infile=haverw.dat
Define a range inside the data range for OUT= dataset,
Using the START=19920101 END=20041231 LIBNAME options.

Obs	DATE	FA	FCM1M	FM1	FTA1MA	FTB3	LICN
1	1992	3466.3	.	965.31	.	3.45415	407.340
2	1993	3624.6	.	1077.69	.	3.01654	344.934
3	1994	3875.8	.	1144.85	.	4.28673	340.054
4	1995	4209.3	.	1142.70	.	5.51058	357.038
5	1996	4399.1	.	1106.46	.	5.02096	351.358
6	1997	4820.3	.	1069.23	.	5.06885	321.513
7	1998	5254.8	.	1079.56	.	4.80726	317.077
8	1999	5608.1	.	1101.14	.	4.66154	301.581
9	2000	6115.4	.	1104.07	.	5.84644	301.108
10	2001	6436.2	2.31368	1136.31	11.753	3.44471	402.583
11	2002	7024.9	1.63115	1192.03	18.798	1.61548	402.796
12	2003	7302.9	1.02346	1268.40	16.089	1.01413	399.137
13	2004	7950.5	1.26642	1337.89	13.019	1.37557	345.109

Example 42.2: Viewing Quarterly Time Series from a Haver Database

The following statements specify a quarterly frequency conversion of all time series for the period spanning April 1, 2001, to December 31, 2004:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=quarterly
        start=20010401
```

```
        end=20041231
        force=freq;

data hwoutq;
   set lib1.haverw;
run;

title1 'Haver Analytics Database, Frequency=quarterly, infile=haverw.dat';
title2 '  Define a range inside the data range for OUT= dataset';
title3 '  Using the START=20010401 END=20041231 LIBNAME options.';

proc print data=hwoutq;
run;
```

The resulting SAS data set hwoutq is shown in Output 42.2.1.

Output 42.2.1 Defining a Range inside the Data Range for Quarterly Time Series

Haver Analytics Database, Frequency=quarterly, infile=haverw.dat
Define a range inside the data range for OUT= dataset
Using the START=20010401 END=20041231 LIBNAME options.

Obs	DATE	FA	FCM1M	FM1	FTA1MA	FTB3	LICN
1	2001Q2	6225.4	.	1115.75	.	3.68308	356.577
2	2001Q3	6425.9	2.98167	1157.90	12.077	3.27615	368.408
3	2001Q4	6436.2	2.00538	1169.62	11.753	1.95308	477.685
4	2002Q1	6396.3	1.73077	1186.92	22.309	1.72615	456.292
5	2002Q2	6563.5	1.72769	1183.30	17.126	1.72077	368.592
6	2002Q3	6780.0	1.69231	1189.89	21.076	1.64769	352.892
7	2002Q4	7024.9	1.37385	1207.80	18.798	1.36731	433.408
8	2003Q1	7054.5	1.17846	1231.41	24.299	1.15269	458.746
9	2003Q2	7319.6	1.08000	1262.24	14.356	1.05654	386.185
10	2003Q3	7238.6	0.92000	1286.21	16.472	0.92885	361.346
11	2003Q4	7302.9	0.91538	1293.76	16.089	0.91846	390.269
12	2004Q1	7637.3	0.90231	1312.43	21.818	0.91308	400.585
13	2004Q2	7769.8	0.94692	1332.75	12.547	1.06885	310.508
14	2004Q3	7949.5	1.34923	1343.79	21.549	1.49393	305.862
15	2004Q4	7950.5	1.82429	1362.60	13.019	2.01731	362.171

Example 42.3: Viewing Monthly Time Series from a Haver Database

The following statements convert weekly time series to a monthly frequency:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=monthly
        start=20040401
        end=20041231
        force=freq;

data hwoutm;
   set lib1.haverw;
run;
```

```
title1 'Haver Analytics Database, Frequency=monthly, infile=haverw.dat';
title2 '  Define a range inside the data range for OUT= dataset';
title3 '  Using the START=20040401 END=20041231 LIBNAME options.';

proc print data=hwoutm;
run;
```

The result from using the range of April 1, 2004, to December 31, 2004, is shown in Output 42.3.1.

Output 42.3.1 Defining a Range inside the Data Range for Monthly Time Series

Haver Analytics Database, Frequency=monthly, infile=haverw.dat
Define a range inside the data range for OUT= dataset
Using the START=20040401 END=20041231 LIBNAME options.

Obs	DATE	FA	FCM1M	FM1	FTA1MA	FTB3	LICN
1	APR2004	7703.8	0.9140	1325.73	16.946	0.93900	317.36
2	MAY2004	7704.7	0.9075	1332.96	25.043	1.03375	297.00
3	JUN2004	7769.8	1.0275	1339.50	12.547	1.26625	315.45
4	JUL2004	7859.5	1.1840	1330.13	21.823	1.34900	357.32
5	AUG2004	7890.0	1.3650	1347.84	25.213	1.48000	276.70
6	SEP2004	7949.5	1.5400	1352.40	21.549	1.65000	270.70
7	OCT2004	7967.6	1.6140	1355.28	21.322	1.74750	304.24
8	NOV2004	8053.4	1.9125	1366.06	21.862	2.05625	335.85
9	DEC2004	7950.5	1.9640	1365.60	13.019	2.20200	441.16

Example 42.4: Viewing Weekly Time Series from a Haver Database

The following statements show weekly data that span from September 1, 2004, to December 31, 2004:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=weekly
        start=20040901
        end=20041231;

data hwoutw;
   set lib1.haverw;
run;

title1 'Haver Analytics Database, Frequency=weekly, infile=haverw.dat';
title2 '  Define a range inside the data range for OUT= dataset';
title3 '  Using the START=20040901 END=20041231 LIBNAME options.';

proc print data=hwoutw;
run;
```

Output 42.4.1 shows the output.

Output 42.4.1 Defining a Range inside the Data Range for Weekly Time Series

Haver Analytics Database, Frequency=weekly, infile=haverw.dat
Define a range inside the data range for OUT= dataset
Using the START=20040901 END=20041231 LIBNAME options.

Obs	DATE	FA	FCM1M	FM1	FTA1MA	FTB3	LICN
1	29AUG2004	7890.0	1.39	1360.8	27.342	1.515	275.2
2	05SEP2004	7906.2	1.46	1353.7	25.213	1.580	273.7
3	12SEP2004	7962.7	1.57	1338.3	25.255	1.635	250.6
4	19SEP2004	7982.1	1.57	1345.6	15.292	1.640	275.8
5	26SEP2004	7987.9	1.56	1359.7	15.068	1.685	282.7
6	03OCT2004	7949.5	1.54	1366.0	21.549	1.710	279.6
7	10OCT2004	7932.4	1.56	1362.3	17.183	1.685	338.7
8	17OCT2004	7956.9	1.59	1350.1	17.438	1.680	279.8
9	24OCT2004	7957.3	1.63	1346.0	12.133	1.770	317.6
10	31OCT2004	7967.6	1.75	1362.7	21.322	1.855	305.5
11	07NOV2004	7954.1	1.84	1350.4	22.028	1.950	354.8
12	14NOV2004	8009.7	1.89	1354.8	25.495	2.045	311.9
13	21NOV2004	7938.3	1.93	1364.5	24.000	2.075	356.0
14	28NOV2004	8053.4	1.99	1381.3	24.424	2.155	320.7
15	05DEC2004	8010.7	2.05	1379.3	21.862	2.195	472.7
16	12DEC2004	8054.8	2.08	1355.1	22.178	2.210	370.6
17	19DEC2004	8019.2	1.98	1358.3	12.066	2.200	374.7
18	26DEC2004	7995.5	1.89	1366.3	12.787	2.180	446.6

Example 42.5: Viewing Daily Time Series from a Haver Database

Consider viewing the Haver Analytics daily database named haverd. The contents of this database can be seen by submitting the following DATA step:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=daily
        start=20041201
        end=20041231;

data hwoutd;
   set lib1.haverd;
run;

title1 'Haver Analytics Database, HAVERD.DAT';
title2 'PROC CONTENTS for Time Series converted to daily frequency';
proc contents data=hwoutd;
run;
```

Output 42.5.1 shows the output of PROC CONTENTS with the time ID variable DATE followed by the time series variables FCM10, FCM1M, FFED, FFP1D, FXAUS, and TCC with their corresponding attributes such as type, length, format, and label.

Output 42.5.1 Examining the Contents of a Daily Haver Analytics Database, haverd.dat

Haver Analytics Database, HAVERD.DAT
PROC CONTENTS for Time Series converted to daily frequency

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	DATE	Num	8	DATE9.	Date of Observation
2	FCM10	Num	8		10-Year Treasury Note Yield at Constant Maturity (Avg, % p.a.)
3	FCM1M	Num	8		1-Month Treasury Bill Market Bid Yield at Constant Maturity (%)
4	FFED	Num	8		Federal Funds [Effective] Rate (% p.a.)
5	FFP1D	Num	8		1-Day AA Financial Commercial Paper (% per annum)
6	FXAUS	Num	8		Foreign Exchange Rate: Australia (US$/Australian$)
7	TCC	Num	8		Treasury: Closing Operating Cash Balance (Today, Mil.$)

Example 42.6: Limiting the Range of Time Series from a Haver Database

The following statements limit the range of data to the month of December:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=daily
        start=20041201
        end=20041231;

data hwoutd;
   set lib1.haverd;
run;

title1 'Haver Analytics Database, Frequency=daily, infile=haverd.dat';
title2 '   Define a range inside the data range for OUT= dataset';
title3 '   Using the START=20041201 END=20041231 LIBNAME options.';

proc print data=hwoutd;
run;
```

Note that Output 42.6.1 for daily conversion shows the frequency as the SAS time interval for WEEKDAY.

Output 42.6.1 Defining a Range inside the Data Range for Daily Time Series

Haver Analytics Database, Frequency=daily, infile=haverd.dat
Define a range inside the data range for OUT= dataset
Using the START=20041201 END=20041231 LIBNAME options.

Obs	DATE	FCM10	FCM1M	FFED	FFP1D	FXAUS	TCC
1	01DEC2004	4.38	2.06	2.04	2.01	0.7754	7564
2	02DEC2004	4.40	2.06	2.00	1.98	0.7769	8502
3	03DEC2004	4.27	2.06	1.98	1.96	0.7778	7405
4	06DEC2004	4.24	2.09	2.04	1.98	0.7748	7019
5	07DEC2004	4.23	2.08	1.99	1.99	0.7754	15520
6	08DEC2004	4.14	2.08	2.01	1.98	0.7545	12329
7	09DEC2004	4.19	2.07	2.05	2.03	0.7532	5441
8	10DEC2004	4.16	2.07	2.09	2.07	0.7495	6368
9	13DEC2004	4.16	2.04	2.18	2.13	0.7592	11395
10	14DEC2004	4.14	2.01	2.24	2.22	0.7566	13695
11	15DEC2004	4.09	1.98	2.31	2.27	0.7652	39765
12	16DEC2004	4.19	1.93	2.26	2.24	0.7563	33640
13	17DEC2004	4.21	1.95	2.23	2.20	0.7607	32764
14	20DEC2004	4.21	1.97	2.26	2.21	0.7644	36216
15	21DEC2004	4.18	1.92	2.24	2.21	0.7660	35056
16	22DEC2004	4.21	1.84	2.25	2.22	0.7656	34599
17	23DEC2004	4.23	1.83	2.34	2.08	0.7654	24467
18	24DEC2004	.	.	2.27	.	0.7689	26898
19	27DEC2004	4.30	1.90	2.24	2.26	0.7777	31874
20	28DEC2004	4.31	1.88	2.24	2.24	0.7787	30513
21	29DEC2004	4.33	1.76	2.23	2.23	0.7709	34754
22	30DEC2004	4.27	1.68	2.24	2.18	0.7785	20045
23	31DEC2004	4.24	1.89	1.97	2.18	0.7805	24690

Example 42.7: Using the WHERE Statement to Subset Time Series from a Haver Database

Using a WHERE statement in the DATA step can be useful for further subsetting.

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=daily start=20041101 end=20041231;

data hwoutd;
   set lib1.haverd;
   where date  between '01nov2004'd and '01dec2004'd;
run;

title1 'Haver Analytics Database, Frequency=daily, infile=haverd.dat';
title2 '   Define a range inside the data range for OUT= dataset';
title3 '   Using the START=20041101 END=20041231 LIBNAME options.';
title4 'Subset further: where date between 01nov2004 and 31dec2004.';
proc print data=hwoutd;
run;
```

Output 42.7.1 shows that the time slice of November 1, 2004, to December 31, 2004, is narrowed further by the DATE test in the WHERE statement to stop at December 1, 2004.

Output 42.7.1 Defining a Range Using the WHERE Statement, START=20041101, and END=20041231

Haver Analytics Database, Frequency=daily, infile=haverd.dat
Define a range inside the data range for OUT= dataset
Using the START=20041101 END=20041231 LIBNAME options.
Subset further: where date between 01nov2004 and 31dec2004.

Obs	DATE	FCM10	FCM1M	FFED	FFP1D	FXAUS	TCC
1	01NOV2004	4.11	1.79	1.83	1.80	0.7460	35111
2	02NOV2004	4.10	1.86	1.74	1.74	0.7447	34091
3	03NOV2004	4.09	1.83	1.73	1.73	0.7539	14862
4	04NOV2004	4.10	1.85	1.77	1.75	0.7585	23304
5	05NOV2004	4.21	1.86	1.76	1.75	0.7620	19872
6	08NOV2004	4.22	1.88	1.80	1.84	0.7578	21095
7	09NOV2004	4.22	1.89	1.79	1.81	0.7618	16390
8	10NOV2004	4.25	1.88	1.92	1.85	0.7592	12872
9	11NOV2004	.	.	1.92	.	.	12872
10	12NOV2004	4.20	1.91	2.02	1.96	0.7685	28926
11	15NOV2004	4.20	1.92	2.06	2.03	0.7719	10480
12	16NOV2004	4.21	1.93	1.98	1.95	0.7728	13417
13	17NOV2004	4.14	1.90	1.99	1.93	0.7833	10506
14	18NOV2004	4.12	1.91	1.99	1.94	0.7786	6293
15	19NOV2004	4.20	1.98	1.99	1.93	0.7852	5100
16	22NOV2004	4.18	1.98	2.01	1.96	0.7839	6045
17	23NOV2004	4.19	1.99	2.00	1.95	0.7860	18135
18	24NOV2004	4.20	1.98	2.02	1.89	0.7863	14109
19	25NOV2004	.	.	2.02	.	.	14109
20	26NOV2004	4.24	2.01	2.01	1.97	0.7903	20588
21	29NOV2004	4.34	2.02	2.03	2.00	0.7852	24322
22	30NOV2004	4.36	2.07	2.02	2.04	0.7723	18033
23	01DEC2004	4.38	2.06	2.04	2.01	0.7754	7564

Example 42.8: Using the KEEP Option to Subset Time Series from a Haver Database

To select specific time series, you can use the KEEP= or DROP= option as follows:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=daily
        start=20041101
        end=20041231
        keep="FCM*";

data hwoutd;
   set lib1.haverd;
run;

title1 'Haver Analytics Database, Frequency=daily, infile=haverd.dat';
title2 '   Define a range inside the data range for OUT= dataset';
title3 '   Using the START=20041101 END=20041231 LIBNAME options.';
title4 '   Subset further: Using keep="FCM*" LIBNAME option ';
proc print data=hwoutd;
run;
```

Output 42.8.1 shows two series that are selected by using KEEP="FCM*" in the LIBNAME statement.

Output 42.8.1 Using the KEEP Option and Defining a Range Using START=20041101 and END=20041231

Haver Analytics Database, Frequency=daily, infile=haverd.dat
Define a range inside the data range for OUT= dataset
Using the START=20041101 END=20041231 LIBNAME options.
Subset further: Using keep="FCM*" LIBNAME option

Obs	DATE	FCM10	FCM1M
1	01NOV2004	4.11	1.79
2	02NOV2004	4.10	1.86
3	03NOV2004	4.09	1.83
4	04NOV2004	4.10	1.85
5	05NOV2004	4.21	1.86
6	08NOV2004	4.22	1.88
7	09NOV2004	4.22	1.89
8	10NOV2004	4.25	1.88
9	11NOV2004	.	.
10	12NOV2004	4.20	1.91
11	15NOV2004	4.20	1.92
12	16NOV2004	4.21	1.93
13	17NOV2004	4.14	1.90
14	18NOV2004	4.12	1.91
15	19NOV2004	4.20	1.98
16	22NOV2004	4.18	1.98
17	23NOV2004	4.19	1.99
18	24NOV2004	4.20	1.98
19	25NOV2004	.	.
20	26NOV2004	4.24	2.01
21	29NOV2004	4.34	2.02
22	30NOV2004	4.36	2.07
23	01DEC2004	4.38	2.06
24	02DEC2004	4.40	2.06
25	03DEC2004	4.27	2.06
26	06DEC2004	4.24	2.09
27	07DEC2004	4.23	2.08
28	08DEC2004	4.14	2.08
29	09DEC2004	4.19	2.07
30	10DEC2004	4.16	2.07
31	13DEC2004	4.16	2.04
32	14DEC2004	4.14	2.01
33	15DEC2004	4.09	1.98
34	16DEC2004	4.19	1.93
35	17DEC2004	4.21	1.95
36	20DEC2004	4.21	1.97
37	21DEC2004	4.18	1.92
38	22DEC2004	4.21	1.84
39	23DEC2004	4.23	1.83
40	24DEC2004	.	.
41	27DEC2004	4.30	1.90
42	28DEC2004	4.31	1.88
43	29DEC2004	4.33	1.76

Output 42.8.1 *continued*

Haver Analytics Database, Frequency=daily, infile=haverd.dat
Define a range inside the data range for OUT= dataset
Using the START=20041101 END=20041231 LIBNAME options.
Subset further: Using keep="FCM*" LIBNAME option

Obs	DATE	FCM10	FCM1M
44	30DEC2004	4.27	1.68
45	31DEC2004	4.24	1.89

You can use the DROP option to drop specific variables from a Haver database. To specify this option, use DROP= instead of KEEP=.

Example 42.9: Using the SOURCE Option to Subset Time Series from a Haver Database

You can use the SOURCE= or DROPSOURCE= option to select specific variables that belong to a certain source, similar to the way you use the KEEP= or DROP= option.

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=daily
        start=20041101
        end=20041223
        source="FRB";

data hwoutd;
   set lib1.haverd;
run;

title1 'Haver Analytics Database, Frequency=daily, infile=haverd.dat';
title2 '   Define a range inside the data range for OUT= dataset';
title3 '   Using the START=20041101 END=20041223 LIBNAME options.';
title4 '   Subset further: Using source="FRB" LIBNAME option';
proc print data=hwoutd;
run;
```

Output 42.9.1 shows two series that are selected by using SOURCE="FRB" in the LIBNAME statement.

Output 42.9.1 Using the SOURCE Option and Defining a Range Using START=20041101 and END=20041223

Haver Analytics Database, Frequency=daily, infile=haverd.dat
Define a range inside the data range for OUT= dataset
Using the START=20041101 END=20041223 LIBNAME options.
Subset further: Using source="FRB" LIBNAME option

Obs	DATE	FCM10	FFED	FFP1D	FXAUS
1	01NOV2004	4.11	1.83	1.80	0.7460
2	02NOV2004	4.10	1.74	1.74	0.7447
3	03NOV2004	4.09	1.73	1.73	0.7539
4	04NOV2004	4.10	1.77	1.75	0.7585
5	05NOV2004	4.21	1.76	1.75	0.7620
6	08NOV2004	4.22	1.80	1.84	0.7578
7	09NOV2004	4.22	1.79	1.81	0.7618
8	10NOV2004	4.25	1.92	1.85	0.7592
9	11NOV2004	.	1.92	.	.
10	12NOV2004	4.20	2.02	1.96	0.7685
11	15NOV2004	4.20	2.06	2.03	0.7719
12	16NOV2004	4.21	1.98	1.95	0.7728
13	17NOV2004	4.14	1.99	1.93	0.7833
14	18NOV2004	4.12	1.99	1.94	0.7786
15	19NOV2004	4.20	1.99	1.93	0.7852
16	22NOV2004	4.18	2.01	1.96	0.7839
17	23NOV2004	4.19	2.00	1.95	0.7860
18	24NOV2004	4.20	2.02	1.89	0.7863
19	25NOV2004	.	2.02	.	.
20	26NOV2004	4.24	2.01	1.97	0.7903
21	29NOV2004	4.34	2.03	2.00	0.7852
22	30NOV2004	4.36	2.02	2.04	0.7723
23	01DEC2004	4.38	2.04	2.01	0.7754
24	02DEC2004	4.40	2.00	1.98	0.7769
25	03DEC2004	4.27	1.98	1.96	0.7778
26	06DEC2004	4.24	2.04	1.98	0.7748
27	07DEC2004	4.23	1.99	1.99	0.7754
28	08DEC2004	4.14	2.01	1.98	0.7545
29	09DEC2004	4.19	2.05	2.03	0.7532
30	10DEC2004	4.16	2.09	2.07	0.7495
31	13DEC2004	4.16	2.18	2.13	0.7592
32	14DEC2004	4.14	2.24	2.22	0.7566
33	15DEC2004	4.09	2.31	2.27	0.7652
34	16DEC2004	4.19	2.26	2.24	0.7563
35	17DEC2004	4.21	2.23	2.20	0.7607
36	20DEC2004	4.21	2.26	2.21	0.7644
37	21DEC2004	4.18	2.24	2.21	0.7660
38	22DEC2004	4.21	2.25	2.22	0.7656
39	23DEC2004	4.23	2.34	2.08	0.7654

Example 42.10: Using the GROUP Option to Subset Time Series from a Haver Database

You can use the GROUP= or DROPGROUP= option to select specific variables that belong to a certain group, similar to the way you use the KEEP= or DROP= option.

Output 42.10.1, Output 42.10.2, and Output 42.10.3 show three different cross sections of the same database, haverw, by specifying three unique GROUP= options: GROUP="F*" in LIBNAME LIB1, GROUP="M*" in LIBNAME LIB2, and GROUP="E*" in LIBNAME LIB3.

The following statements specify GROUP="F*" in the LIBNAME LIB1 statement:

```
libname lib1 sasehavr "%sysget(HAVER_DATA)"
        freq=week.6
        force=freq
        start=20040102
        end=20041001
        group="F*";

data hwoutwA;
   set lib1.haverw;
run;

title1 'Haver Analytics Database, Frequency=week.6, infile=haverw.dat';
title2 '   Define a range inside the data range for OUT= dataset';
title3 '   Using the START=20040102 END=20041001 LIBNAME options.';
title4 '   Subset further: Using group="F*" LIBNAME option';
proc print data=hwoutwA;
run;
```

Output 42.10.1 shows the output.

Output 42.10.1 Using the GROUP=F* Option and Defining a Range

Haver Analytics Database, Frequency=week.6, infile=haverw.dat
Define a range inside the data range for OUT= dataset
Using the START=20040102 END=20041001 LIBNAME options.
Subset further: Using group="F*" LIBNAME option

Obs	DATE	FCM1M	FTA1MA	FTB3
1	01JAN2004	0.86	16.089	0.885
2	08JAN2004	0.88	12.757	0.920
3	15JAN2004	0.84	12.141	0.870
4	22JAN2004	0.79	12.593	0.875
5	29JAN2004	0.86	17.357	0.890
6	05FEB2004	0.90	21.759	0.920
7	12FEB2004	0.90	21.557	0.920
8	19FEB2004	0.92	21.580	0.915
9	26FEB2004	0.96	21.390	0.930
10	04MAR2004	0.97	24.119	0.940
11	11MAR2004	0.96	24.294	0.930
12	18MAR2004	0.94	23.334	0.945
13	25MAR2004	0.95	21.400	0.930
14	01APR2004	0.95	21.818	0.945
15	08APR2004	0.94	17.255	0.930
16	15APR2004	0.92	14.143	0.915
17	22APR2004	0.89	14.136	0.935
18	29APR2004	0.87	16.946	0.970
19	06MAY2004	0.89	22.772	0.985
20	13MAY2004	0.89	23.113	1.060
21	20MAY2004	0.91	25.407	1.040
22	27MAY2004	0.94	25.043	1.050
23	03JUN2004	0.97	27.847	1.130
24	10JUN2004	1.01	27.240	1.230
25	17JUN2004	1.05	17.969	1.390
26	24JUN2004	1.08	12.159	1.315
27	01JUL2004	1.11	12.547	1.355
28	08JUL2004	1.14	21.303	1.320
29	15JUL2004	1.16	25.024	1.315
30	22JUL2004	1.21	25.327	1.330
31	29JUL2004	1.30	21.823	1.425
32	05AUG2004	1.34	21.631	1.465
33	12AUG2004	1.37	28.237	1.470
34	19AUG2004	1.36	26.070	1.470
35	26AUG2004	1.39	27.342	1.515
36	02SEP2004	1.46	25.213	1.580
37	09SEP2004	1.57	25.255	1.635
38	16SEP2004	1.57	15.292	1.640
39	23SEP2004	1.56	15.068	1.685
40	30SEP2004	1.54	21.549	1.710

The following statements specify GROUP="M*" in the LIBNAME LIB2 statement:

```
libname lib2 sasehavr "%sysget(HAVER_DATA)"
        freq=week.6
        force=freq start=20040102
        end=20041001
        group="M*";

data hwoutwB;
   set lib2.haverw;
run;

title1 'Haver Analytics Database, Frequency=week.6, infile=haverw.dat';
title2 '   Define a range inside the data range for OUT= dataset';
title3 '   Using the START=20040102 END=20041001 LIBNAME options.';
title4 '   Subset further: Using group="M*" LIBNAME option';
proc print data=hwoutwB;
run;
```

Output 42.10.2 shows the output.

Output 42.10.2 Using the GROUP=M* Option and Defining a Range

Haver Analytics Database, Frequency=week.6, infile=haverw.dat
Define a range inside the data range for OUT= dataset
Using the START=20040102 END=20041001 LIBNAME options.
Subset further: Using group="M*" LIBNAME option

Obs	DATE	FA	FM1
1	31DEC2003	7302.9	1298.2
2	07JAN2004	7351.2	1294.3
3	14JAN2004	7378.5	1286.8
4	21JAN2004	7434.7	1296.7
5	28JAN2004	7492.4	1305.1
6	04FEB2004	7510.4	1303.1
7	11FEB2004	7577.8	1309.1
8	18FEB2004	7648.7	1317.0
9	25FEB2004	7530.6	1321.1
10	03MAR2004	7546.7	1316.2
11	10MAR2004	7602.0	1312.7
12	17MAR2004	7603.0	1324.0
13	24MAR2004	7625.5	1337.6
14	31MAR2004	7637.3	1337.9
15	07APR2004	7667.4	1327.3
16	14APR2004	7692.5	1321.8
17	21APR2004	7698.4	1322.2
18	28APR2004	7703.8	1331.6
19	05MAY2004	7686.8	1342.5
20	12MAY2004	7734.6	1325.5
21	19MAY2004	7695.8	1330.1
22	26MAY2004	7704.7	1337.7
23	02JUN2004	7715.1	1329.0
24	09JUN2004	7754.0	1324.4
25	16JUN2004	7753.2	1336.4
26	23JUN2004	7796.2	1345.8
27	30JUN2004	7769.8	1351.4
28	07JUL2004	7852.3	1330.1
29	14JUL2004	7852.8	1326.3
30	21JUL2004	7854.7	1323.5
31	28JUL2004	7859.5	1340.6
32	04AUG2004	7847.9	1337.3
33	11AUG2004	7888.7	1340.1
34	18AUG2004	7851.8	1347.3
35	25AUG2004	7890.0	1360.8
36	01SEP2004	7906.2	1353.7
37	08SEP2004	7962.7	1338.3
38	15SEP2004	7982.1	1345.6
39	22SEP2004	7987.9	1359.7
40	29SEP2004	7949.5	1366.0

The following statements specify GROUP="E*" in the LIBNAME LIB3 statement:

```
libname lib3 sasehavr "%sysget(HAVER_DATA)"
        freq=week.6
        force=freq
        start=20040102
        end=20041001
        group="E*";

data hwoutwC;
   set lib3.haverw;
run;

title1 'Haver Analytics Database, Frequency=week.6, infile=haverw.dat';
title2 '   Define a range inside the data range for OUT= dataset';
title3 '   Using the START=20040102 END=20041001 LIBNAME options.';
title4 '   Subset further: Using group="E*" LIBNAME option';
proc print data=hwoutwC;
run;
```

Output 42.10.3 shows the output.

Output 42.10.3 Using the GROUP=E* Option and Defining a Range

Haver Analytics Database, Frequency=week.6, infile=haverw.dat
Define a range inside the data range for OUT= dataset
Using the START=20040102 END=20041001 LIBNAME options.
Subset further: Using group="E*" LIBNAME option

Obs	DATE	LICN
1	02JAN2004	552.8
2	09JAN2004	677.9
3	16JAN2004	490.8
4	23JAN2004	382.3
5	30JAN2004	406.3
6	06FEB2004	433.2
7	13FEB2004	341.6
8	20FEB2004	328.2
9	27FEB2004	342.1
10	05MAR2004	339.0
11	12MAR2004	312.1
12	19MAR2004	304.5
13	26MAR2004	296.8
14	02APR2004	304.2
15	09APR2004	350.7
16	16APR2004	335.0
17	23APR2004	313.7
18	30APR2004	283.2
19	07MAY2004	292.8
20	14MAY2004	297.1
21	21MAY2004	294.0
22	28MAY2004	304.1
23	04JUN2004	308.2
24	11JUN2004	312.4
25	18JUN2004	322.5
26	25JUN2004	318.7
27	02JUL2004	349.9
28	09JUL2004	444.5
29	16JUL2004	394.4
30	23JUL2004	315.7
31	30JUL2004	282.1
32	06AUG2004	291.5
33	13AUG2004	268.0
34	20AUG2004	272.1
35	27AUG2004	275.2
36	03SEP2004	273.7
37	10SEP2004	250.6
38	17SEP2004	275.8
39	24SEP2004	282.7
40	01OCT2004	279.6

Example 42.11: Using the OUTSELECT=ON Option to View the Key Selection Variables in a Haver Database

Suppose you want to select your time series based on geography codes or source codes. To construct your wildcard for selection, first run with the OUTSELECT=ON option to see the possible values for each selection key.

```
Libname lib1 sasehavr "%sysget(HAVER_DATA)"
        outselect=on ;

data validD1;
   set lib1.haverd;
run;

title1 'OUTSELECT=ON, Print the OUT= Data Set';
title2 'Shows the Values for Key Selection Variables:';
title3 'Name, Source, Geog1, Geog2, Shortsrc, Longsrc';
title4 'OUTSELECT=ON, the CONTENTS Procedure with Variable Names and Labels';
proc print data=validD1;
run;

proc contents data=validD1;
run;
```

Output 42.11.1 shows the output values for each key selection variable.

Output 42.11.1 OUTSELECT=ON Option Shows the Values for Key Selection Variables

OUTSELECT=ON, Print the OUT= Data Set
Shows the Values for Key Selection Variables:
Name, Source, Geog1, Geog2, Shortsrc, Longsrc
OUTSELECT=ON, the CONTENTS Procedure with Variable Names and Labels

Obs	NAME	SOURCE	GEOG1	GEOG2	SHORTSRC	LONGSRC	FCM10	FCM1M	FFED	FFP1D	FXAUS	TCC
1	NAME	SOURCE	GEOG1	GEOG2	SHORTSRC	LONGSRC						
2	FCM10	FRB	0000000		FRB	Federal Reserve Board						
3	FCM1M	UST	0000000		FRB	Federal Reserve Board						
4	FFED	FRB	0000000		FRB	Federal Reserve Board						
5	FFP1D	FRB	0000000		FRB	Federal Reserve Board						
6	FXAUS	FRB	0000000		FRBNY	Federal Reserve Bank of New York						
7	TCC	UST	0000000		TREASURY	U.S. Treasury						

If you also want to see a list of all the variables and their corresponding labels for this OUTSELECT=ON data set, you can run the CONTENTS Procedure.

Output 42.11.2 shows the contents of the output data set.

Output 42.11.2 OUTSELECT=ON Option Shows the Contents of HAVERD.DAT

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Label
7	FCM10	Char	8	10-Year Treasury Note Yield at Constant Maturity (Avg, % p.a.)
8	FCM1M	Char	8	1-Month Treasury Bill Market Bid Yield at Constant Maturity (%)
9	FFED	Char	8	Federal Funds [Effective] Rate (% p.a.)
10	FFP1D	Char	8	1-Day AA Financial Commercial Paper (% per annum)
11	FXAUS	Char	8	Foreign Exchange Rate: Australia (US$/Australian$)
3	GEOG1	Char	8	DLXRECORD.Geography1
4	GEOG2	Char	8	DLXRECORD.Geography2
6	LONGSRC	Char	70	DLXRECORD.LongSource
1	NAME	Char	10	DLXRECORD.VarName
5	SHORTSRC	Char	10	DLXRECORD.ShortSourc
2	SOURCE	Char	6	DLXRECORD.Source
12	TCC	Char	8	Treasury: Closing Operating Cash Balance (Today, Mil.$)

Example 42.12: Selecting Variables Based on Short Source Key Code

Using the information from Example 42.11, you can now select time series by using selection keys such as the SHORT= , GEOG1=, or GEOG2= options. Since the short source values are nontrivial in database haverd, it is best in this case to use the SHORT= option. For more information about using geography codes as selection keys, see Output 42.13.1 for the GEOG1= option and Output 42.13.2 for the GEOG2= option.

```
Libname lib1 sasehavr "%sysget(HAVER_DATA)"
        short="GOLDMAN, FRB, CRB";
data validE2;
   set lib1.haverd;
   where date  between '18jan2005'd and '29mar2005'd;
run;

title1 'SHORT= option list: GOLDMAN, FRB, CRB';
title2 'Should contain these time series:';
title3 'FCM10, FCM1M, FFED, FFP1D';
title4 'SHORT= option, Print the OUT= ValidE2 Data Set';
proc print data=validE2;
run;

title4 'SHORT= option, Print the Contents of OUT= ValidE2 Data Set';
proc contents data=validE2;
run;
```

Output 42.12.1 shows the output for the SHORT= option.

Output 42.12.1 SHORT= Option Shows the Selected Variables

SHORT= option list: GOLDMAN, FRB, CRB
Should contain these time series:
FCM10, FCM1M, FFED, FFP1D
SHORT= option, Print the OUT= ValidE2 Data Set

Obs	DATE	FCM10	FCM1M	FFED	FFP1D
1	18JAN2005	4.21	2.05	2.31	2.30
2	19JAN2005	4.20	1.95	2.19	2.22
3	20JAN2005	4.17	1.89	2.25	2.22
4	21JAN2005	4.16	2.02	2.26	2.19
5	24JAN2005	4.14	2.05	2.26	2.22
6	25JAN2005	4.20	2.13	2.29	2.22
7	26JAN2005	4.21	2.16	2.33	2.26
8	27JAN2005	4.22	2.16	2.39	2.30
9	28JAN2005	4.16	2.12	2.48	2.37
10	31JAN2005	4.14	2.06	2.50	2.47
11	01FEB2005	4.15	2.23	2.40	2.47
12	02FEB2005	4.15	2.22	2.29	2.45
13	03FEB2005	4.18	2.18	2.49	2.46
14	04FEB2005	4.09	2.20	2.51	2.45
15	07FEB2005	4.07	2.27	2.50	2.47
16	08FEB2005	4.05	2.34	2.48	2.45
17	09FEB2005	4.00	2.34	2.50	2.45
18	10FEB2005	4.07	2.35	2.51	2.47
19	11FEB2005	4.10	2.36	2.50	2.48
20	14FEB2005	4.08	2.37	2.51	2.50
21	15FEB2005	4.10	2.40	2.53	2.54
22	16FEB2005	4.16	2.39	2.48	2.45
23	17FEB2005	4.19	2.40	2.50	2.47
24	18FEB2005	4.27	2.39	2.51	2.45
25	21FEB2005	.	.	2.51	.
26	22FEB2005	4.29	2.43	2.57	2.49
27	23FEB2005	4.27	2.47	2.53	2.48
28	24FEB2005	4.29	2.48	2.55	2.52
29	25FEB2005	4.27	2.50	2.54	2.52
30	28FEB2005	4.36	2.51	2.52	2.58
31	01MAR2005	4.38	2.55	2.39	2.51
32	02MAR2005	4.38	2.54	2.48	2.44
33	03MAR2005	4.39	2.55	2.51	2.49
34	04MAR2005	4.32	2.56	2.50	2.46
35	07MAR2005	4.31	2.59	2.51	2.49
36	08MAR2005	4.38	2.61	2.49	2.47
37	09MAR2005	4.52	2.60	2.50	2.45
38	10MAR2005	4.48	2.60	2.52	2.49
39	11MAR2005	4.56	2.60	2.51	2.48
40	14MAR2005	4.52	2.62	2.59	2.53
41	15MAR2005	4.54	2.70	2.61	2.60
42	16MAR2005	4.52	2.68	2.57	2.50
43	17MAR2005	4.47	2.68	2.68	2.58
44	18MAR2005	4.51	2.70	2.70	2.68

Output 42.12.1 *continued*

SHORT= option list: GOLDMAN, FRB, CRB
Should contain these time series:
FCM10, FCM1M, FFED, FFP1D
SHORT= option, Print the OUT= ValidE2 Data Set

Obs	DATE	FCM10	FCM1M	FFED	FFP1D
45	21MAR2005	4.53	2.72	2.71	2.72
46	22MAR2005	4.63	2.77	2.72	2.68
47	23MAR2005	4.61	2.72	2.73	2.69
48	24MAR2005	4.60	2.70	2.75	2.62
49	25MAR2005	.	.	2.80	2.59
50	28MAR2005	4.64	2.69	2.79	2.79
51	29MAR2005	.	.	.	2.76

If you also want to see a list of all the variables and their corresponding labels for this data set, you can run the CONTENTS Procedure.

Output 42.12.2 shows the output.

Output 42.12.2 SHORT= Option Shows the Contents of the validE2 Data Set

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	DATE	Num	8	DATE9.	Date of Observation
2	FCM10	Num	8		10-Year Treasury Note Yield at Constant Maturity (Avg, % p.a.)
3	FCM1M	Num	8		1-Month Treasury Bill Market Bid Yield at Constant Maturity (%)
4	FFED	Num	8		Federal Funds [Effective] Rate (% p.a.)
5	FFP1D	Num	8		1-Day AA Financial Commercial Paper (% per annum)

Example 42.13: Selecting Variables Based on Geography Key Codes

Since the haverd database did not have interesting geography codes, the following statements access the INTWKLY database by using its more complete geography key codes to select the desired time series from the specified geography codes:

```
Libname lib1 sasehavr "%sysget(HAVER_DATA_NEW)"
        outselect=on
   keep="R273RF3,X924USBE,R023DF,R273G1,F023A,F158FBS,F023ACR,X156VEB,F023ACE";

data valid1(keep=NAME SOURCE GEOG1 GEOG2 SHORTSRC LONGSRC);
   set lib1.intwkly;
run;

title1 'OUTSELECT=ON, Print the OUT= Data Set';
title2 'Shows the Values for Key Selection Variables:';
title3 'Name, Source, Geog1, Geog2, Shortsrc, Longsrc';
title4 'OUTSELECT=ON, the CONTENTS Procedure with Variable Names and Labels';
```

```
proc print data=valid1;
run;

Libname lib2 sasehavr "%sysget(HAVER_DATA_NEW)"
   geog1="156";

data valid2(
   keep=date R273RF3 X924USBE R023DF R273G1 F023A F158FBS F023ACR X156VEB F023ACE);
   set lib2.intwkly;
run;

title1 'Only one GEOG1 Code, 156, contains time series X156VEB';
title2 'Select Geography Code 1 Option:';
title3 'GEOG1= option';
title4 'Only Time Series X156VEB has Geog1 = 156';

proc contents
   data=valid2;
run;

Libname lib3 sasehavr "%sysget(HAVER_DATA_NEW)"
   geog2="299";

data valid3(
   keep=date R273RF3 X924USBE R023DF R273G1 F023A F158FBS F023ACR X156VEB F023ACE);
   set lib3.intwkly;
run;

title1 'Only one GEOG2 Code, 299, contains time series X156VEB';
title2 'Select Geography Code 2 Option:';
title3 'GEOG2= option';
title4 'Only Time Series X156VEB has Geog2 = 299';

proc contents
   data=valid3;
run;

title1 'Compare GEOG1 Code 156';
title2 'Over nonmissing values range';
title3 'With GEOG2 Code 299';
title4 'Over nonmissing values range';

proc compare listall briefsummary criterion=1.0e-5
   base=valid2(
   where=( date between '09jan1998'd and '28dec2007'd ))
   compare=valid3(
   where=( date between '09jan1998'd and '28dec2007'd ));
run;
```

Output 42.13.1, Output 42.13.2, Output 42.13.3, and Output 42.13.4 show the output.

Output 42.13.1 OUTSELECT=ON Option Shows the Values for Key Selection Variables

OUTSELECT=ON, Print the OUT= Data Set
Shows the Values for Key Selection Variables:
Name, Source, Geog1, Geog2, Shortsrc, Longsrc
OUTSELECT=ON, the CONTENTS Procedure with Variable Names and Labels

Obs	NAME	SOURCE	GEOG1	GEOG2	SHORTSRC	LONGSRC
1	NAME	SOURCE	GEOG1	GEOG2	SHORTSRC	LONGSRC
2	F023A	STLF	023		ECB	European Central Bank
3	F023ACE	STLF	023		ECB	European Central Bank
4	F023ACR	STLF	023		ECB	European Central Bank
5	F158FBS	---	158		JMoF	Ministry of Finance
6	R023DF	---	023		ECB	European Central Bank
7	X156VEB	STLF	156	299	BOCAN	Bank of Canada
8	X924USBE	STLF	924	111	SAFE	China State Administration of Foreign Exchange

Output 42.13.2 Only One GEOG1 Code, 156, Contains Time Series X156VEB

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	DATE	Num	8	DATE9.	Date of Observation
2	X156VEB	Num	8		Canada: Venezuelan Bolivar Noon Exchange Rate (C$/Bolivar)

Output 42.13.3 Only One GEOG2 Code, 299, Contains Time Series X156VEB

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	DATE	Num	8	DATE9.	Date of Observation
2	X156VEB	Num	8		Canada: Venezuelan Bolivar Noon Exchange Rate (C$/Bolivar)

Output 42.13.4 Comparing GEOG1 and GEOG2 Access of INTWKLY Haver DLX Database

OUTSELECT=ON, Print the OUT= Data Set
Shows the Values for Key Selection Variables:
Name, Source, Geog1, Geog2, Shortsrc, Longsrc
OUTSELECT=ON, the CONTENTS Procedure with Variable Names and Labels

Obs	NAME	SOURCE	GEOG1	GEOG2	SHORTSRC	LONGSRC
1	NAME	SOURCE	GEOG1	GEOG2	SHORTSRC	LONGSRC
2	F023A	STLF	023		ECB	European Central Bank
3	F023ACE	STLF	023		ECB	European Central Bank
4	F023ACR	STLF	023		ECB	European Central Bank
5	F158FBS	---	158		JMoF	Ministry of Finance
6	R023DF	---	023		ECB	European Central Bank
7	X156VEB	STLF	156	299	BOCAN	Bank of Canada
8	X924USBE	STLF	924	111	SAFE	China State Administration of Foreign Exchange

Output 42.13.4 *continued*

Only one GEOG1 Code, 156, contains time series X156VEB
Select Geography Code 1 Option:
GEOG1= option
Only Time Series X156VEB has Geog1 = 156

The CONTENTS Procedure

Data Set Name	WORK.VALID2	**Observations**	2404
Member Type	DATA	**Variables**	2
Engine	V9	**Indexes**	0
Created	03/17/2014 17:32:43	**Observation Length**	16
Last Modified	03/17/2014 17:32:43	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information

Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	4062
Obs in First Data Page	2404
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	C:\Users\saskff\AppData\Local\Temp\SAS Temporary Files_TD2172_D79286_\valid2.sas7bdat
Release Created	9.0401M2
Host Created	W32_7PRO

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	DATE	Num	8	DATE9.	Date of Observation
2	X156VEB	Num	8		Canada: Venezuelan Bolivar Noon Exchange Rate (C$/Bolivar)

Output 42.13.4 *continued*

Only one GEOG2 Code, 299, contains time series X156VEB
Select Geography Code 2 Option:
GEOG2= option
Only Time Series X156VEB has Geog2 = 299

The CONTENTS Procedure

Data Set Name	WORK.VALID3	**Observations**	682
Member Type	DATA	**Variables**	2
Engine	V9	**Indexes**	0
Created	03/17/2014 17:39:03	**Observation Length**	16
Last Modified	03/17/2014 17:39:03	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information

Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	4062
Obs in First Data Page	682
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	C:\Users\saskff\AppData\Local\Temp\SAS Temporary Files_TD2172_D79286_\valid3.sas7bdat
Release Created	9.0401M2
Host Created	W32_7PRO

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	DATE	Num	8	DATE9.	Date of Observation
2	X156VEB	Num	8		Canada: Venezuelan Bolivar Noon Exchange Rate (C$/Bolivar)

Compare GEOG1 Code 156
Over nonmissing values range
With GEOG2 Code 299
Over nonmissing values range

```
The COMPARE Procedure
Comparison of WORK.VALID2 with WORK.VALID3
(Method=RELATIVE(2.22E-09), Criterion=0.00001)

NOTE: No unequal values were found. All values compared are exactly equal.
```

References

Haver Analytics (2010), *DLX API Programmer's Reference*, New York.

Haver Analytics (2012), *DLX Database Profile*, New York.
http://www.haver.com/databaseprofiles.html

Haver Analytics (2012), *Data Link Express*, New York.
http://www.haver.com/datalink.html

Subject Index

Syntax Index