SAS/ETS® 15.1 User's Guide
The SASEQUAN Interface Engine

This document is an individual chapter from *SAS/ETS® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/ETS® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/ETS® 15.1 User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

Chapter 52
The SASEQUAN Interface Engine

Contents

Overview: SASEQUAN Interface Engine

The SASEQUAN interface engine enables SAS users to retrieve economic and other time series data from the Quandl website, which is hosted by Quandl. The Quandl website offers access to 8 million time series data sets from 400 sources in finance, economics, society, health, energy, demography, and more. These time series are updated at annual, quarterly, monthly, weekly, and daily intervals. The time series on the Quandl website contain observation or measurement periods that are associated with data values. Although the Quandl API also supports data tables, the SASEQUAN engine does not support data tables at this time.

The SASEQUAN interface engine uses the LIBNAME statement to enable you to specify how to subset your Quandl data and how to collapse the selected time series to the same update 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 SASEQUAN interface engine supports Linux X64 (LAX) and Windows. Although the SASEQUAN engine uses the Quandl API (default is version 3), it is not endorsed or certified by Quandl. By using the SASEQUAN interface engine, you are agreeing to comply with the Quandl terms of use, which are described on the web page at the following URL: **`https://www.quandl.com/about/terms`**.

Getting Started: SASEQUAN Interface Engine

You can query the Quandl data set to retrieve the observations or data values for a list of time series by specifying the Quandl code of the data set. The Quandl code consists of a source code and a table code for the data set that contains the time series that you want to read into SAS. You must also specify your unique Quandl API key (authentication token for unlimited access). To obtain your own unique API key, visit the Quandl website at the following URL: **`https://www.quandl.com/users/sign_up`**.

The Quandl API key is a 20-character mixed-case alphanumeric string, such as "abCDefghiJKLMn123456," and is represented by 'XXXXXXXXXXXXXXXXXXXX' in the APIKEY= option in the following example. In addition, the example URLs in this section and in the section "Examples: SASEQUAN Interface Engine" on page 3717 use the same Quandl API key as the argument *your_quan_apikey*.

After you have your assigned Quandl API key and have agreed to the Quandl terms of use, you are almost ready to download Quandl data. Before you download, make sure you have the necessary rights to work with the data.

Now that your are informed about the terms of use of the Quandl data, you can use your Quandl API key to access the Quandl data, as shown in the following example.

In this example, and the ones that follow in "Examples: SASEQUAN Interface Engine" on page 3717, use the SAS option SSLCALISTLOC=<specify the location of your CA certificates here>. The specification that is shown as SSLCALISTLOC= "/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem" is for demonstration purposes only. Specify your own location of your trusted certificates inside the double quotes.

The statements that follow enable you to access oil prices from the National Stock Exchange of India's time series data from September 1, 2013, to November 5, 2013, on a daily basis. The observations are sorted by the time ID variable DATE. The output is shown in Output 52.1.

```
options validvarname=any
   sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";
title 'Retrieve Data for Oil India Limited Prices';
libname _all_ clear;

libname quan sasequan "/sasusr/playpens/saskff/quan/test/"
   OUTXML=oiltd
   XMLMAP="/sasusr/playpens/saskff/quan/test/oiltd.map"
   APIKEY='XXXXXXXXXXXXXXXXXXX'
   IDLIST='NSE/OIL';

data oil_gsa;
   set quan.oiltd;
run;

proc contents data=oil_gsa; run;
proc print data=oil_gsa(firstobs=1328 obs=1342); run;
```

Figure 52.1 Oil India Limited Prices: Oil_Gsa (FIRSTOBS=1328 OBS=1342)

Retrieve Data for Oil India Limited Prices

Obs	date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
1328	2015-02-02	536.20	540.90	530.25	534.00	533.25	201704	1077.25
1329	2015-02-03	539.80	541.00	526.25	531.50	531.35	923694	4910.35
1330	2015-02-04	541.00	550.45	536.40	545.50	548.75	485793	2644.40
1331	2015-02-05	548.85	549.00	538.25	540.50	540.05	877473	4742.75
1332	2015-02-06	536.50	552.90	536.50	545.35	547.00	358329	1962.28
1333	2015-02-09	545.00	553.75	530.00	540.00	543.00	608323	3332.38
1334	2015-02-10	540.00	546.45	527.00	531.45	530.85	326785	1759.67
1335	2015-02-11	532.00	536.40	529.10	530.30	530.95	116276	618.56
1336	2015-02-12	534.65	536.00	528.00	531.95	531.65	189407	1006.99
1337	2015-02-13	521.00	525.90	495.10	504.00	500.20	895268	4542.81
1338	2015-02-16	505.00	513.90	495.00	495.00	499.00	379163	1909.42
1339	2015-02-18	501.80	506.50	494.40	500.95	501.10	261958	1314.47
1340	2015-02-19	503.30	506.00	494.15	497.00	497.30	161816	806.24
1341	2015-02-20	499.00	502.90	493.00	494.30	494.40	220134	1092.32
1342	2015-02-23	500.00	500.00	485.20	487.80	487.30	194121	952.37

The XML data that the Quandl website returns are placed in a file that is named by the OUTXML= option—in this case, *OILTD1.xml*. Note that the SASEQUAN engine appends a numeral to the XML file name, and the file extension (.xml) is excluded from the file name that appears in the OUTXML= option. This XML data file resides in the current working directory. These data are read into a SAS data set in the folder location that is given inside the string enclosed in double quotation marks in the SASEQUAN LIBNAME statement. So, in the preceding example, if the QUANDL environment variable is set to

```
/sasusr/playpens/saskff/quan/test/
```

Then the SAS data set (created when the XML file is read into SAS) is located at

```
/sasusr/playpens/saskff/quan/test/OIL_GSA.sas7bdat
```

An equivalent LIBNAME statement that does not use any environment variables could be as follows:

```
libname quan sasequan "/sasusr/playpens/saskff/fame940/ets/test/"
   OUTXML=oiltd
   XMLMAP="/sasusr/playpens/saskff/quan/test/oiltd.map"
   APIKEY='XXXXXXXXXXXXXXXXXXXX'
   IDLIST='NSE/OIL';
```

You could also use either a SAS macro variable or a system environment variable to store the value of your Quandl API key so that the key does not appear explicitly in your SAS code. The XML map that is created is assigned the full path name that the XMLMAP= option specifies. The SASEQUAN engine appends a numeral to the XML file name to indicate the position of the Quandl code in the IDLIST= option.

The IDLIST= option specifies the list of Quandl data sets (that contain time series) that you want to retrieve. This option accepts a string, enclosed in single quotation marks, that denotes a list of one or more Quandl data sets that you select (keep) in the resulting SAS data set. The result, OILTD, is named in the DATA step and is shown in Figure 52.1. The preceding example uses only one Quandl code, which is in the first position of the IDLIST= option, so the numeral 1 is appended to the name of the XML file, resulting in OILTD1.xml.

It is more efficient to use the DATA step to store your Quandl data in a SAS data set and then refer to the SAS data set directly in your PROC PRINT or PROC SGPLOT statement. You can also refer to the SASEQUAN libref directly, as in the statement

```
proc print data=quan.oiltd; run;
```

This statement uses the member name, OILTD, in the PROC PRINT statement; this usage corresponds to specifying the OUTXML=OILTD option. Although using this statement might seem easier, it is not as efficient, because every time you use the SASEQUAN libref, the Quandl interface engine reads the entire XML file into SAS again. So it is better to refer to the SAS data set repeatedly than to invoke the interface engine repeatedly.

Syntax: SASEQUAN Interface Engine

The SASEQUAN interface engine uses standard engine syntax to read the observations or data values for one or more Quandl data sets that can contain one or more time series in each data set. Table 52.1 summarizes the options that the SASEQUAN engine uses. In addition, there is one required option: API_KEY='*quan_api_key*'.

Table 52.1 Summary of LIBNAME *libref* SASEQUAN Options

Option	Description
APIKEY=	Specifies the required Quandl access key that enables you to access the data that the Quandl website provides
AUTOMAP=	Specifies whether or not to overwrite the existing XML map file
COLLAPSE=	Specifies the reporting frequency (lower frequency to collapse the output results to). The valid reporting frequencies are daily, weekly, monthly, quarterly, annual, and none.
COLUMN=	Specifies one column (time series) to keep in the output results. The rest of the columns are dropped from the output results. When more than one ID is specified in the IDLIST= option, the specified column index is kept for each ID.
CONNECT=	Specifies whether or not to use the connect method for a secure connection via a proxy server. You must specify the PROXY= option when you use the CONNECT=ON option. See the PROXY= option.
DEBUG=	Specifies whether or not to include diagnostic message logging in the SAS log window
END=	Specifies the end date (trim_end) for the observation period (`'YYYY-MM-DD'` formatted string, optional; the default is 1776-07-04 (earliest available))
FORMAT=	Specifies a file extension that indicates the type of file to retrieve. Only XML is supported for the SASEQUAN interface engine.
FREQ=	Specifies the frequency of the selected data: daily, weekly, monthly, quarterly, or annual. When the IDLIST= option contains more than one Quandl code, the FREQ= option aggregates higher-frequency data series to lower-frequency time series (such as converting a monthly time series to an annual time series).
IDLIST=	Specifies a list of Quandl codes for Quandl data set codes for accessing Quandl time series data. To select more than one data set, list the unique Quandl codes, separated by commas. There is a limit of nine Quandl codes in the IDLIST= option.
MAPREF=	Specifies the fileref used for the map file assignment
OUTXML=	Specifies the name of the output SAS data set and the XML file(s) requested by the IDLIST= option. When more than one time series ID is listed in the IDLIST= option, the SASEQUAN engine appends the positional integer (1 f or the first time series ID, 2 for the second time series ID, and so on) to the name specified by the OUTXML= option.
PROXY=	Specifies the proxy server that you want to use (if you have trouble connecting without specifying a proxy). If you also need the connect method for a secure connection, use the CONNECT=ON option in addition to the PROXY= option. See the CONNECT= option.
ROWS=	Specifies the maximum number of observations (rows) to return (integer between 1 and 100,000, optional; the default is 100,000)
SORT=	Specifies the order of the results in ascending or descending observation-date order. The valid sort arguments are *asc* and *desc*; the default is *asc*.
START=	Specifies the start date (trim_start) for the observation period (`'YYYY-MM-DD'` formatted string, optional; the default is 9999-12-31 (latest available))
TRANS=	Specifies the transformation method to be used for data transformation. The valid transformation arguments are DIFF, RDIFF, RDIFF_FROM, CUMUL, NORMALIZE, and NONE; the default is NONE. See Table 52.2 for formulas.
XMLMAP=	Specifies the fully qualified name of the location where the XMLmap file is automatically stored. By default, XMLMAP=Quan.map.

The LIBNAME *libref* SASEQUAN Statement

LIBNAME *libref* **SASEQUAN** '*physical-name*' *options* **;**

The LIBNAME statement assigns a SAS library reference (libref) to the physical path of the directory where the SAS data set is stored that contains the downloaded Quandl data. The required *physical-name* argument specifies the location of the folder where your SAS data set resides. It should end with a backslash if you are in a Windows environment and a forward slash if you are in a UNIX environment.

You can specify the following *options* in the LIBNAME *libref* SASEQUAN statement.

APIKEY='*quan_apikey***'**
specifies the Quandl authentication token or access key that enables you to access the data that the Quandl website provides. The Quandl access key is a 20-character mixed-case alphanumeric string, and it is required. It must be enclosed in single quotation marks. You can request your *quan_apikey* by visiting the website at the following URL:

`https://www.quandl.com/users/sign_up`

AUTOMAP=REPLACE | REUSE
specifies whether or not to overwrite the existing XML map file. You can specify the following values:

REPLACE specifies that the XML map file be overwritten, and ensures that the most current XML map that is generated by the SASEQUAN engine and named by the XMLMAP= option is used.

REUSE specifies that the XML map file not be overwritten, and ensures that a pre-existing XML map file that is named by the XMLMAP= option is used.

By default, AUTOMAP=REPLACE.

COLLAPSE=DAILY | WEEKLY | MONTHLY | QUARTERLY | ANNUAL | NONE
specifies the frequency to which you want to collapse the reporting frequency. You can specify the following values:

DAILY collapses the report to a daily frequency.

WEEKLY collapses the report to a weekly frequency.

MONTHLY collapses the report to a monthly frequency.

QUARTERLY collapses the report to a quarterly frequency.

ANNUAL collapses the report to an annual frequency.

NONE does not collapse the report.

This option is not required. By default, COLLAPSE=NONE when IDLIST=option specifies one Quandl code, but when the IDLIST= option specifies more than one Quandl code, the default for the collapse frequency is set to the same frequency that is specified in the FREQ= option.

The Quandl frequency-collapsing feature reports the native (higher-frequency) time series at a lower frequency (the collapse frequency). When you collapse the frequency of a data set, Quandl returns the last observation for the given period. So if you collapse a daily data set to monthly, you get a

sample of the original data set in which the observation for each month is the last data point available for that month. When you specify more than one Quandl code in the IDLIST= option, it is important to check that the *from* date and *to* date of every selected series use the same fiscal year, so that the reporting interval of the merged date values from all the data sets aligns to the same date for the first observation in the range. For example, if multiple Quandl codes are listed in the IDLIST= option, some annual time series have *from* dates that start in January, and some annual time series have *from* dates that start in June, then the merged data set will have observation dates reported for both January and June (if COLLAPSE=NONE), resulting in a semiannual interval instead of an annual interval in the merged data. To preserve the annual frequency, specify COLLAPSE=ANNUAL so that each annual time series aligns with the appropriate annual date in the merged data set. The COLLAPSE= option is applied to each Quandl data set that is specified in the IDLIST= option, so that when the data sets are merged, the reporting frequency is equal to the COLLAPSE= frequency. The resulting merged SAS data set contains the same data as the Quandl "supersets" that were created from the same Quandl codes in the IDLIST= option. Although Quandl supersets are no longer supported by Quandl, newer Quandl API methods are available for merging multiple time series by using the Quandl Excel Add-In. The SASEQUAN interface engine uses the Quandl data sets API to request each time series in the IDLIST= option, enabling you to seamlessly store the merged time series in one SAS data set. For more information about the various available methods for Quandl data access, see the web page at following URL: **`https://www.quandl.com/docs/api#data-organization`**.

NOTE: The COLLAPSE=MONTHLY option reports the daily, weekly, and monthly native frequencies of the time series at a monthly frequency (the collapse frequency). If you specify an annual native frequency time series in the IDLIST= option, then it will not be selected when COLLAPSE=MONTHLY is specified. Only the time series that have native frequencies higher than the reporting frequency specified in the COLLAPSE= option are selected.

NOTE: It is highly recommended that you use the COLLAPSE= option when you specify more than one Quandl code in the IDLIST= option.

CAUTION: If the COLLAPSE=NONE option is specified, then undesirable time intervals can occur when you specify more than one Quandl code in the IDLIST= option.

COLUMN=*quan_column_index*

specifies the column index that you want to keep in the output results. Specify only one column index, and it will be applied to each Quandl code (ID) that is specified in the IDLIST= option. For example, if there are three columns of data, you can specify COLUMN=1 to keep the first column, COLUMN=2 to keep the second column, or COLUMN=3 to keep the third column.

CONNECT=ON | OFF

specifies whether or not to use the connect method along with the PROXY= option. **NOTE:** You must use the PROXY= option and specify your proxy server in addition to the CONNECT=ON option when you want to use the connect method. For more information about secure connections, see the PROXY= option.

DEBUG=ON | OFF

specifies whether or not to include diagnostic message logging in the SAS log window. This information can be very useful for troubleshooting a problem.

END='*quan_enddate*'
specifies the end date for the time series in the format `'YYYY-MM-DD'`. This option is not required, and the default is 9999-12-31 (latest available). The date must be enclosed in single quotation marks.

FORMAT=XML
specifies the format of the file to be received from the Quandl website. Although Quandl can report data in many formats, the SASEQUAN engine supports only the XML format.

FREQ=DAILY | WEEKLY | MONTHLY | QUARTERLY | ANNUAL
specifies a lower frequency to aggregate values to. The FREQ= option also selects only those time series that aggregate to the specified frequency. In Quandl data, the highest frequency is daily, and the lowest frequency is annual. You can specify the following values:

DAILY	selects time series that aggregate to a daily frequency.
WEEKLY	selects time series that aggregate to a weekly frequency.
MONTHLY	selects time series that aggregate to a monthly frequency.
QUARTERLY	selects time series that aggregate to a quarterly frequency.
ANNUAL	selects time series that aggregate to an annual frequency.

The FREQ= option is not required, and the default value is the native frequency of the Quandl data set.

NOTE: An error is returned if you specify a frequency higher than the native frequency of the selected series. For example, if a series has the native frequency "Annual," it is not possible to aggregate the series to the higher "Monthly" frequency. To find the native frequency of a time series, enter the time series' Quandl code (in the database_code and dataset_code fields) in the following URL in your web browser:

`https://www.quandl.com/api/v3/datasets/database_code/dataset_code/data.xml`

The output gives you the time series data along with its native frequency, which is given in the "Frequency" field.

NOTE: When you specify a single Quandl code in the IDLIST= option and the FREQ= option is not specified or is an empty string, the native frequency of the time series in that data set is used as the reporting frequency unless you specify the reporting frequency in the COLLAPSE= option. When you specify multiple data sets (and time series) in the IDLIST= option, the "Annual" frequency is used as the default frequency unless you specify the reporting frequency in the COLLAPSE= option. If any time series in the IDLIST= option have a lower native frequency than the specified frequency, then those time series are dropped from the list and excluded from the output.

IDLIST='*quan_idlist*'
specifies the list of Quandl codes for the data sets that contain the time series to be included in the output SAS data set. There is a limit of nine Quandl codes in the IDLIST= option. This list is comma-delimited and must be enclosed in single quotation marks.

MAPREF=*quan_xmlmapref*
specifies the fileref to use for the map assignment. For an example of the SASEQUAN engine that uses the MAPREF= and XMLMAP= options in the FILENAME statement in order to assign a file name, as in the following, see the section "Getting Started: SASEQUAN Interface Engine" on page 3706:

```
FILENAME MyMap "/sasusr/playpens/saskff/quan/test/oiltd.map";
```

You can use the MAPREF= and XMLMAP= options to control where the map resides, what you name the map, and how you refer to it with a fileref. You can use the OUTXML= option to name your XML data file. It is placed in the current working directory. For more information, see the section "SAS OUTXML File" on page 3716. The SET statement (see the section "Getting Started: SASEQUAN Interface Engine" on page 3706) reads observations from the input data set OILTD and stores them in a SAS data set named OIL_GSA.

OUTXML=*quan_xmlfile*

specifies the name of the file where the XML data that are returned from the Quandl website are stored. Each Quandl code that is listed in the IDLIST= option is given a positional numeral: 1 for the first code in the IDLIST, 2 for the second code in the IDLIST, and so on. The engine appends this numeral to the file name of the XML of each data set that the website returns. When all the XML files are retrieved, the data are merged into a SAS data set. When only one Quandl code is used in the IDLIST= option, the file name has the numeral 1 appended to the OUTXML file name. By default, OUTXML=QUAN, which creates a file named *QUAN1.xml* in the current working directory. The SAS data set that is created when the XML data are read into SAS is placed in the folder specified by the physical path in the LIBNAME libref SASEQUAN statement.

PROXY="*quan_proxyserver***"**

specifies which proxy server to use. This option is not required. The specified proxy server is used only when a connection-refused error or a connection-timed-out error occurs. For *quan_proxyserver*, specify the server's HTTP address followed by a colon and the port number, and enclose that string in double quotation marks; for example, PROXY="http://inetgw.unx.sas.com:8118". See also the CONNECT= option.

ROWS=*quan_rows*

LIMIT=*quan_rows*

specifies the maximum number of rows (time series observations) to return, which is an integer between 1 and 100,000. This option is not required. By default, ROWS=100000.

SORT=ASC | DESC

ORDER=ASC | DESC

specifies the order in which to sort the date of time series observations. You can specify the following values:

ASC sorts time series observations in ascending date order.

DESC sorts time series observations in descending date order.

This option is not required. By default, SORT=ASC.

START='*quan_startdate***'**

specifies the start date for the time series in the format `'YYYY-MM-DD'`. This option is not required, and the default is 1776-07-04 (earliest available). The date must be enclosed in single quotation marks.

TRANS=CUMUL | DIFF | NORMALIZE | RDIFF | RDIFF_FROM | NONE
TRANSFORMATION=CUMUL | DIFF | NORMALIZE | RDIFF | RDIFF_FROM | NONE
specifies the data value transformation. You can specify the following values:

CUMUL	performs the cumulative function.
DIFF	performs the difference function.
NORMALIZE	performs the normalize function.
RDIFF	performs the ratio difference function.
RDIFF_FROM	gives the latest (nearest to the end date) value as a percentage increment.
NONE	performs no transformation on the data.

This option is not required. By default, TRANS=NONE. The details of the arguments and the corresponding function formulas are presented in Table 52.2.

Table 52.2 Quandl Transformation Codes

Trans Code	Description	Formula
cumul	Cumulative sum	$x_t + x_{t-1} + \cdots + x_{t-N}$
diff	Row-on-row change	$x_t - x_{t-1}$
normalize	Scale series to start at 100	$(\frac{x_t}{x_{t-N}}) \times 100$
rdiff	Row-on-row percentage change	$(\frac{x_t - x_{t-1}}{x_{t-1}})$
rdiff_from	Latest value as percentage increment	$(\frac{x_{\text{latest}} - x_t}{x_t})$

x_t is the value of series x at time period t. N is the number of observations per year, which differs by frequency: Daily ($N = 260$), Annual ($N = 1$), Monthly ($N = 12$), Quarterly ($N = 4$), and Weekly ($N = 52$).

XMLMAP=*quan_xmlmapfile*
specifies the fully qualified name of the location where the XML map file is automatically stored.

Details: SASEQUAN Interface Engine

The SASEQUAN interface engine enables SAS users to access time series data that are stored in Quandl data sets that the Quandl website provides. Every Quandl data set is identified by a unique ID. For example, the Prague Stock Index is uniquely identified by the code PRAGUESE/PX, which you can view by visiting the website at the following URL:

```
https://www.quandl.com/data/PRAGUESE/PX-Prague-Stock-Index-PX
```

The unique code for any data set is always visible on the data set page, next to the words "Quandl Code."

Quandl API Key

The API key that is used in these examples, abCDefghiJKLMn123456, is for demonstration purposes only. To successfully download data from the Quandl website, use your own Quandl API key, which is a 20-character mixed-case alphanumeric string. You can request your own API key by visiting the website at the following URL:

https://www.quandl.com/users/sign_up

Available Sources That Provide Quandl Economic Time Series Data

To obtain a list of the available sources of Quandl economic data, visit the website at the following URL:

https://blog.quandl.com/api-for-economic-data

Useful Lists for Easy Downloading of Quandl Time Series Data

You can use the Quandl data browser to get a list of Quandl codes for the available time series for a specific database. Enter the following URL in your web browser and click on the category or the particular link for that source:

https://www.quandl.com/search?query=

For example, to find the Quandl codes for the Dow Jones Industrial Average Index, you can enter the following URL in your web browser:

https://www.quandl.com/search?query=dow%20jones%20industrial%20average%20index

To see only the free databases, select the free filter in the browser box. The 19 free databases are listed along with each time series Quandl codes.

Available Time Series for Each Quandl Code

To download all the data set codes and data set names available in the FRED (Federal Reserve Economic Data) database, enter the following URL in your web browser:

https://www.quandl.com/api/v3/databases/FRED/codes?api_key=YOURAPIKEY

SAS Output Data Set

You can use a SAS DATA step to write the selected Quandl data to a SAS data set. This enables you to use SAS software to easily analyze the data. If you specify the name of the output data set in the DATA statement, the engine supervisor creates a SAS data set that has the specified name in either the SAS Work library or, if specified, the SAS User library.

The contents of the SAS data set include the date of each observation and the series name of each series that is read from the Quandl data source.

The SASEQUAN interface engine maintains the sort order, so the time series are sorted in the resulting SAS data set by the order that is specified in the SORT= option, by date (time ID), and by variable (time series item name).

You can use the PRINT and CONTENTS procedures to print your output data set and its contents. Alternatively, you can view your SAS output observations by opening the desired output data set in a SAS Explorer window. You can also use the SQL procedure along with your SASEQUAN libref to create a custom view of your data.

SAS OUTXML File

The SAS XML (XML format) data that are returned from the Quandl website are placed in a file that is named by the OUTXML= option. The SASEQUAN interface engine creates a separate XML file for each Quandl code that is listed in the IDLIST= option. The engine numbers each data set's XML file in the order in which it appears in the IDLIST= option, so the first data set has a 1 concatenated to the file name, the second data set has a 2 concatenated to the file name, and so on. In instances of the IDLIST= option that contain more than one Quandl code, the variable names also have the same numeral concatenated to them. This naming convention enables the engine to merge all the selected time series into one SAS data set while preserving the identity of each time series. The SAS XML data file is placed in the current working directory, but the SAS data set (created by reading the XML data into SAS) is placed in the folder specified by the *physical-name* in the LIBNAME *libref* SASEQUAN statement, which is described in the section "The LIBNAME *libref* SASEQUAN Statement" on page 3710.

SAS XML Map File

The XML map that (by default) is automatically created is assigned the full path name that is given by the XMLMAP= option in your LIBNAME *libref* SASEQUAN statement. The map file is either reused (not overwritten) if you specify AUTOMAP=REUSE or overwritten by a new map if you specify AUTOMAP=REPLACE (the default). The SASEQUAN interface engine invokes the XMLV2 engine to create the map and to read the data into SAS.

Examples: SASEQUAN Interface Engine

Example 52.1: Retrieving Historical Price Data for Oil India Limited

This example shows how to use one Quandl code, NSE/OIL, to retrieve historical prices for Oil India Limited, starting September 1, 2013, and ending November 5, 2013, with a daily frequency. The output is shown in Output 52.1.1.

```
options validvarname=any
   sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";

title 'Historical Prices for Oil India Limited';
libname _all_ clear;
libname mylib "/sasusr/playpens/saskff/quan/doc/";

/* export QUANDL=/sasusr/playpens/saskff/quan/test/ */

libname myQoil sasequan "/sasusr/playpens/saskff/quan/test/"
   apikey='XXXXXXXXXXXXXXXXXXX'
   idlist='NSE/OIL'
   format=XML
   outXml=oil
   automap=replace
   mapref=MyMap
   xmlmap="/sasusr/playpens/saskff/quan/test/oil.map"
   start='2013-09-01'
   end='2013-11-05'
   freq='daily'
   collapse='daily'
   ;

data mylib.oilall;
   set myQoil.oil;
run;

proc contents data=mylib.oilall; run;
proc print data=mylib.oilall; run;
```

Output 52.1.1 Historical Prices for Oil India Limited

Historical Prices for Oil India Limited

Obs	date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
1	2013-09-02	435.95	441.65	427.20	431.00	431.45	174437	755.45
2	2013-09-03	439.90	439.90	427.00	428.50	429.05	199749	860.41
3	2013-09-04	435.00	435.00	426.15	429.50	429.45	790295	3396.42
4	2013-09-05	430.00	439.95	430.00	435.00	432.60	586678	2539.29
5	2013-09-06	437.00	450.00	433.30	445.25	445.15	543652	2402.79
6	2013-09-10	450.00	465.00	446.10	462.10	460.65	663553	2997.61
7	2013-09-11	462.00	485.00	461.00	466.00	466.70	371647	1733.05
8	2013-09-12	458.05	466.00	446.10	448.70	448.10	211533	959.45
9	2013-09-13	452.50	484.00	448.15	471.05	470.25	826546	3884.01
10	2013-09-16	483.70	484.00	458.80	476.00	467.00	335598	1593.84
11	2013-09-17	467.00	479.20	460.35	473.00	475.55	241830	1148.25
12	2013-09-18	471.20	481.85	471.20	480.00	479.70	182343	868.29
13	2013-09-19	485.00	499.00	476.00	491.10	493.75	457626	2236.70
14	2013-09-20	493.00	493.00	459.00	472.15	466.50	295333	1393.19
15	2013-09-23	466.75	487.00	464.00	480.00	480.40	273803	1302.58
16	2013-09-24	481.90	481.90	464.10	466.00	465.80	314456	1486.22
17	2013-09-25	467.90	473.30	466.10	470.15	470.35	738597	3472.11
18	2013-09-26	471.00	473.70	447.30	453.00	451.95	537088	2434.72
19	2013-09-27	456.70	462.00	450.10	452.00	454.30	345246	1571.16
20	2013-09-30	449.70	457.80	435.00	435.25	437.40	394564	1742.00
21	2013-10-01	437.15	449.35	432.00	449.00	447.90	308033	1367.86
22	2013-10-03	448.00	461.00	444.15	457.10	458.90	197974	898.93
23	2013-10-04	456.95	464.00	455.55	461.50	461.10	227214	1047.43
24	2013-10-07	464.90	471.45	450.00	468.00	464.40	240571	1098.48
25	2013-10-08	467.00	471.65	461.00	463.00	462.25	208627	964.45
26	2013-10-09	462.00	465.80	456.75	465.50	465.10	101852	472.35
27	2013-10-10	465.10	468.50	459.20	460.30	462.25	339738	1578.62
28	2013-10-11	465.00	468.70	457.00	467.50	463.25	213591	983.10
29	2013-10-14	464.65	467.90	461.00	464.10	463.95	125129	580.40
30	2013-10-15	464.00	471.80	456.55	459.30	460.55	407231	1877.01
31	2013-10-17	460.50	465.00	452.50	453.20	454.40	220366	1009.36
32	2013-10-18	457.00	465.95	457.00	465.00	464.55	185891	857.04
33	2013-10-21	465.00	471.90	458.70	468.00	468.85	114130	531.62
34	2013-10-22	468.85	473.20	461.15	465.70	466.65	198435	924.12
35	2013-10-23	463.05	469.50	451.40	456.00	457.65	469852	2152.30
36	2013-10-24	458.00	462.95	452.00	452.00	453.40	246085	1126.66
37	2013-10-25	458.00	460.05	450.00	454.00	454.65	272926	1238.47
38	2013-10-28	455.00	459.70	445.10	457.00	454.10	173547	785.17
39	2013-10-29	457.00	469.30	451.50	464.00	459.95	258106	1179.18
40	2013-10-30	460.20	467.80	453.95	463.50	463.25	301971	1391.67
41	2013-10-31	463.00	481.00	456.00	473.00	473.85	472301	2221.88
42	2013-11-01	470.10	481.00	464.50	480.00	475.05	318091	1495.83
43	2013-11-03	479.00	482.20	475.25	476.00	477.70	34250	163.85
44	2013-11-05	475.05	476.90	465.10	467.05	469.35	190319	894.87

Example 52.2: Retrieving Data by Using Three Quandl Codes

This example shows how to use three Quandl codes of different native frequencies to retrieve quarterly data for corporate profits after tax (FRED/CP), gross domestic product (FRED/GDP), and total consumer credit owned and securitized, outstanding (TOTALSL). The output is shown in Output 52.2.1.

```
title 'Retrieve Data for Three Time Series: FRED/CP, FRED/GDP, FRED/TOTALSL';
libname _all_ clear;
options validvarname=any
   sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";

libname mylib "/sasusr/playpens/saskff/quan/doc/";

/* export QUANDL=/sasusr/playpens/saskff/quan/test/ */

libname myQ3 sasequan "/sasusr/playpens/saskff/quan/test/"
   OUTXML=fred3
   AUTOMAP=replace
   MAPREF=MyMap
   XMLMAP="/sasusr/playpens/saskff/quan/test/fred3.map"
   APIKEY='XXXXXXXXXXXXXXXXXXXX'
   IDLIST='FRED/CP,FRED/GDP,FRED/TOTALSL'
   FORMAT=xml
   START='2009-07-01'
   END='2013-07-01'
   FREQ='quarterly'
   COLLAPSE='quarterly'
   ;

data mylib.thrall;
   set myQ3.fred3;
   label Value_1 = "Corporate Profits After Tax";
   label Value_2 = "Gross Domestic Product, 1 Decimal";
   label Value_3 = "Total Consumer Credit Owned and Securitized, Outstanding";
run;

proc contents data=mylib.thrall; run;
proc print data=mylib.thrall label; run;
```

Output 52.2.1 Retrieve Data for Corporate Profits after Tax, Gross Domestic Product, Total Consumer Credit Owned and Securitized, Outstanding

Retrieve Data for Three Time Series: FRED/CP, FRED/GDP, FRED/TOTALSL

Obs	date	Corporate Profits After Tax	Gross Domestic Product, 1 Decimal	Total Consumer Credit Owned and Securitized, Outstanding
1	2009-09-30	1346.76	14420.31	2573.28
2	2009-12-31	1462.69	14628.02	2555.02
3	2010-03-31	1530.36	14721.35	2536.75
4	2010-06-30	1516.85	14926.10	2520.00
5	2010-09-30	1599.90	15079.92	2519.34
6	2010-12-31	1598.83	15240.84	2646.81
7	2011-03-31	1463.90	15285.83	2672.78
8	2011-06-30	1528.83	15496.19	2694.73
9	2011-09-30	1539.11	15591.85	2719.13
10	2011-12-31	1616.77	15796.46	2757.07
11	2012-03-31	1879.81	16019.76	2791.08
12	2012-06-30	1801.44	16152.26	2835.89
13	2012-09-30	1818.05	16257.15	2871.41
14	2012-12-31	1785.65	16358.86	2918.26
15	2013-03-31	1766.30	16569.59	2965.75
16	2013-06-30	1757.43	16637.93	3001.79
17	2013-09-30	1792.61	16848.75	3016.12

Example 52.3: Retrieving Data for the NIKKEI JASDAQ Stock Average Index

This example shows how to use one Quandl code, NIKKEI/JASDAQ, to retrieve the price data for the Nikkei JASDAQ Stock Average Index, which is adjusted by 'Dow adjustment' starting June 6, 2018, and ending September 5, 2018, with a daily native frequency. The output is shown in Output 52.3.1.

```
options validvarname=any
   sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";

title 'NIKKEI JASDAQ Stock Average Index ';
libname _all_ clear;
libname mylib "/sasusr/playpens/saskff/quan/doc/";

/* export QUANDL=/sasusr/playpens/saskff/quan/test/ */

libname myTOP20 sasequan "/sasusr/playpens/saskff/quan/test/"
   apikey='XXXXXXXXXXXXXXXXXXX'
   idlist='NIKKEI/JASDAQ'
   format=XML
   outXml=jasdaq
   automap=replace
   mapref=MyMap
   xmlmap="/sasusr/playpens/saskff/quan/test/jasdaq.map"
```

```
   start='2018-06-06'
   end='2018-09-05'
   ;

data mylib.jasdaq;
   set myTOP20.jasdaq;
run;

proc contents data=mylib.jasdaq; run;
proc print data=mylib.jasdaq(obs=35); run;
```

Output 52.3.1 Nikkei JASDAQ Stock Average Index

NIKKEI JASDAQ Stock Average Index

Obs	date	Close	Open	High	Low
1	2018-06-06	3907.41	3913.23	3917.36	3905.34
2	2018-06-07	3933.74	3914.68	3935.19	3914.68
3	2018-06-08	3941.61	3934.10	3943.73	3933.06
4	2018-06-11	3952.23	3948.19	3954.03	3938.48
5	2018-06-12	3973.26	3959.92	3975.52	3959.09
6	2018-06-13	3986.03	3977.64	3992.52	3976.86
7	2018-06-14	3981.57	3986.62	3992.63	3980.35
8	2018-06-15	3982.55	3987.93	3993.61	3980.99
9	2018-06-18	3961.87	3987.07	3987.64	3961.83
10	2018-06-19	3915.70	3962.23	3966.99	3910.45
11	2018-06-20	3920.83	3907.19	3922.85	3873.82
12	2018-06-21	3937.68	3924.71	3941.55	3923.49
13	2018-06-22	3915.78	3922.79	3923.16	3911.95
14	2018-06-25	3879.36	3925.20	3933.50	3879.36
15	2018-06-26	3863.49	3856.32	3865.59	3844.95
16	2018-06-27	3864.89	3860.42	3873.45	3855.71
17	2018-06-28	3841.51	3861.01	3862.19	3835.74
18	2018-06-29	3858.22	3849.27	3861.69	3847.24
19	2018-07-02	3826.76	3870.04	3873.97	3826.76
20	2018-07-03	3793.89	3834.64	3852.70	3786.55
21	2018-07-04	3756.89	3783.17	3783.17	3756.89
22	2018-07-05	3679.18	3751.35	3755.01	3679.18
23	2018-07-06	3726.65	3690.47	3729.66	3689.75
24	2018-07-09	3786.09	3740.70	3787.65	3740.56
25	2018-07-10	3800.95	3803.56	3812.56	3799.99
26	2018-07-11	3773.61	3793.23	3793.23	3766.19
27	2018-07-12	3785.08	3775.21	3787.75	3772.72
28	2018-07-13	3793.95	3794.26	3799.79	3790.64
29	2018-07-17	3783.91	3798.85	3798.85	3782.67
30	2018-07-18	3811.59	3795.29	3814.14	3794.90
31	2018-07-19	3808.67	3815.16	3816.76	3806.12
32	2018-07-20	3804.74	3807.20	3816.57	3801.28
33	2018-07-23	3808.96	3802.41	3809.87	3800.50
34	2018-07-24	3827.92	3819.47	3830.40	3816.96
35	2018-07-25	3847.30	3842.70	3849.16	3838.82

Example 52.4: Collapsing Data for the Nikkei JASDAQ Stock Average Index

This example shows how to collapse daily data to a weekly interval by using the same Quandl code as in Example 52.3, NIKKEI/JASDAQ, to retrieve the price data for the Nikkei JASDAQ Stock Average Index, starting June 6, 2018, and ending September 5, 2018, with a daily native frequency. You collapse the data to a weekly frequency by using the COLLAPSE= option. The output is shown in Output 52.4.1.

```
options validvarname=any
   sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";

title 'NIKKEI JASDAQ Stock Average index, COLLAPSE=WEEKLY Option';
libname _all_ clear;
libname mylib "/sasusr/playpens/saskff/quan/doc/";

/* export QUANDL=/sasusr/playpens/saskff/quan/test/ */

libname myTOP20 sasequan "/sasusr/playpens/saskff/quan/test/"
   apikey='XXXXXXXXXXXXXXXXXXX'
   idlist='NIKKEI/JASDAQ'
   format=XML
   outXml=jasdaqW
   automap=replace
   mapref=MyMap
   xmlmap="/sasusr/playpens/saskff/quan/test/jasdaqw.map"
   start='2018-06-06'
   end='2018-09-05'
   collapse=weekly
   ;

data mylib.jasdaqW;
   set myTOP20.jasdaqW;
run;

proc contents data=mylib.jasdaqW; run;
proc print data=mylib.jasdaqW; run;
```

Output 52.4.1 Nikkei JASDAQ Stock Average Index, with COLLAPSE=WEEKLY

NIKKEI JASDAQ Stock Average index, COLLAPSE=WEEKLY Option

Obs	date	Close	Open	High	Low
1	2018-06-10	3941.61	3934.10	3943.73	3933.06
2	2018-06-17	3982.55	3987.93	3993.61	3980.99
3	2018-06-24	3915.78	3922.79	3923.16	3911.95
4	2018-07-01	3858.22	3849.27	3861.69	3847.24
5	2018-07-08	3726.65	3690.47	3729.66	3689.75
6	2018-07-15	3793.95	3794.26	3799.79	3790.64
7	2018-07-22	3804.74	3807.20	3816.57	3801.28
8	2018-07-29	3848.39	3852.38	3854.14	3845.81
9	2018-08-05	3788.41	3813.30	3815.53	3788.41
10	2018-08-12	3745.27	3761.36	3763.16	3743.09
11	2018-08-19	3701.70	3700.83	3707.66	3697.72
12	2018-08-26	3742.90	3734.55	3744.46	3732.05
13	2018-09-02	3823.35	3806.85	3825.29	3804.67
14	2018-09-09	3778.37	3792.62	3800.86	3778.37

Example 52.5: Transforming Data for the Nikkei JASDAQ Stock Average Index

This example shows how to transform daily data by using the DIFF transformation and the same Quandl code as in Example 52.3 and Example 52.4, NIKKEI/JASDAQ, to retrieve the price data for the Nikkei JASDAQ Stock Average Index, starting June 6, 2018, and ending September 5, 2018, with a daily native frequency. You collapse the data to a weekly frequency by using the COLLAPSE= option and perform a transformation by using the TRANS= option. Specify a range by using START='2018-06-06' and END='2018-09-05', a collapse frequency by using COLLAPSE=WEEKLY, and a transformation function by using TRANS=DIFF. The output is shown on Output 52.5.1.

```
options validvarname=any
   sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";

title 'NIKKEI JASDAQ Stock Average Index, TRANS=DIFF Option';
libname _all_ clear;
libname mylib "/sasusr/playpens/saskff/quan/doc/";

/* export QUANDL=/sasusr/playpens/saskff/quan/test/ */

libname myTOP20 sasequan "/sasusr/playpens/saskff/quan/test/"
   apikey='XXXXXXXXXXXXXXXXXXXX'
   idlist='NIKKEI/JASDAQ'
   format=XML
   outXml=jasdaqX
   automap=replace
   mapref=MyMap
   xmlmap="/sasusr/playpens/saskff/quan/test/jasdaqX.map"
   start='2018-06-06'
   end='2018-09-05'
   collapse=weekly
```

```
   trans=diff
   ;

data mylib.jasdaqX;
   set myTOP20.jasdaqX;
run;

proc contents data=mylib.jasdaqX; run;
proc print data=mylib.jasdaqX; run;
```

Output 52.5.1 Nikkei JASDAQ Stock Average Index, Weekly Data with TRANS=DIFF

NIKKEI JASDAQ Stock Average Index, TRANS=DIFF Option

Obs	date	Close	Open	High	Low
1	2018-06-17	40.94	53.83	49.88	47.93
2	2018-06-24	-66.77	-65.14	-70.45	-69.04
3	2018-07-01	-57.56	-73.52	-61.47	-64.71
4	2018-07-08	-131.57	-158.80	-132.03	-157.49
5	2018-07-15	67.30	103.79	70.13	100.89
6	2018-07-22	10.79	12.94	16.78	10.64
7	2018-07-29	43.65	45.18	37.57	44.53
8	2018-08-05	-59.98	-39.08	-38.61	-57.40
9	2018-08-12	-43.14	-51.94	-52.37	-45.32
10	2018-08-19	-43.57	-60.53	-55.50	-45.37
11	2018-08-26	41.20	33.72	36.80	34.33
12	2018-09-02	80.45	72.30	80.83	72.62
13	2018-09-09	-44.98	-14.23	-24.43	-26.30

Example 52.6: Reading Data from Multiple Quandl Data Sets to Merge Multiple Time Series

This example shows how to read data from three Quandl data sets by using the Quandl codes EIA/ELEC_PLANT_CONS_EG_BTU_57692_ALL_ALL_M, BUNDESBANK/BBK01_WT5511, and YALE/SPCOMP to retrieve oil, gold, and S&P Composite prices, dividends, and earnings data. There are eleven time series (one for fuel, one for gold, and nine for the S&P Composite), taken from three different Quandl data sets: DOE/RWTC, BUNDESBANK/BBK01_WT5511, and YAHOO/INDEX_GSPC, respectively. Because the Fuel, Gold, and S&P Composite columns are all from monthly native frequency data sets, you can use the "Annual" collapse frequency to minimize the missing values in the output. Specify a range by using START='2010-12-31' and END='2016-12-31', and specify a collapse frequency by using COLLAPSE=ANNUAL. The output is shown in Output 52.6.1.

```
options validvarname=any
   sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";

title 'Fuels Used for Generating Electricity, Gold, and S&P Composite Stock
       Time Series';
libname _all_ clear;
libname mylib "/sasusr/playpens/saskff/quan/doc/";

/* export QUANDL=/sasusr/playpens/saskff/quan/test/ */

libname mysup sasequan "/sasusr/playpens/saskff/quan/test/"
   apikey='XXXXXXXXXXXXXXXXXXX'
   idlist='EIA/ELEC_PLANT_CONS_EG_BTU_57692_ALL_ALL_M,BUNDESBANK/BBK01_WT5511,YALE/SPCOMP'
   format=XML
   outXml=Tsupe
   automap=replace
   mapref=MyMap
   xmlmap="/sasusr/playpens/saskff/quan/test/Tsupe.map"
   start='2010-12-31'
   end='2016-12-31'
   collapse=annual
   ;

data mylib.Tsupe;
   set mysup.Tsupe;
   label Value_1 = "Electric Fuel Consumption";
   label Value_2= "Gold Price (USD)";
   label 'S&P Composite_3'n= "S&P Composite from SPCOMP";
   label Dividend_3= "Dividend (SPCOMP)";
   label Earnings_3= "Earnings (SPCOMP)";
   label CPI_3= "CPI (SPCOMP)";
   label 'Long Interest Rate_3'n= "Long Interest Rate (SPCOMP)";
   label 'Real Price_3'n= "Real Price (SPCOMP)";
   label 'Real Dividend_3'n= "Real Dividend (SPCOMP)";
   label 'Real Earnings_3'n= "Real Earnings (SPCOMP)";
   label 'Cyclically Adjusted PE Ratio_3'n= "Cyclically Adjusted PE Ratio (SPCOMP)";
run;

proc contents data=mylib.Tsupe; run;
proc print data=mylib.Tsupe label; run;
```

Output 52.6.1 Reading from Multiple Quandl Data Sets: Fuel, Gold, and S&P Composite Data Using COLLAPSE= Option

Fuels Used for Generating Electricity, Gold, and S&P Composite Stock Time Series

Obs	date	Electric Fuel Consumption	Gold Price (USD)	S&P Composite from SPCOMP	Dividend (SPCOMP)	Earnings (SPCOMP)	CPI (SPCOMP)	Long Interest Rate (SPCOMP)
1	2010-12-31	0	1410.25	1241.53	22.73	77.35	219.179	3.29
2	2011-12-31	764	1574.50	1243.32	26.43	86.95	225.672	1.98
3	2012-12-31	1472	1664.00	1422.29	31.25	86.51	229.601	1.72
4	2013-12-31	1042	1201.50	1807.78	34.99	100.20	233.049	2.90
5	2014-12-31	688	1199.25	2054.27	39.44	102.31	234.812	2.21
6	2015-12-31	1398	1062.25	2054.08	43.39	86.53	236.525	2.24
7	2016-12-31	1425	1237.70	2246.63	45.70	94.55	241.432	2.49

Obs	Real Price (SPCOMP)	Real Dividend (SPCOMP)	Real Earnings (SPCOMP)	Cyclically Adjusted PE Ratio (SPCOMP)
1	1429.08	26.1638	89.035	22.3964
2	1389.97	29.5474	97.206	20.5236
3	1562.84	34.3381	95.059	21.2383
4	1957.03	37.8788	108.473	24.8619
5	2207.18	42.3756	109.925	26.7941
6	2190.99	46.2820	92.297	25.9654
7	2347.67	47.7552	98.802	27.8651

References

Federal Reserve Bank of St. Louis (2012). "Economic Research." Accessed November 7, 2012. https://research.stlouisfed.org/.

Kamel, T. (2014a). "Data Sources." Accessed March 14, 2014. https://www.quandl.com/search.

Kamel, T. (2014b). "Useful Lists." Accessed March 14, 2014. https://blog.quandl.com/.

Kamel, T. (2014c). "Using the Quandl API." Accessed March 14, 2014. https://www.quandl.com/help/api/.

Shiller, R. J. (2018). "Online Data." Accessed September 6, 2018. http://www.econ.yale.edu/~shiller/data.htm.

Subject Index

Syntax Index