THE POWER TO KNOW.

SAS/ETS® 14.2 User's Guide
The SASEQUAN Interface Engine

This document is an individual chapter from *SAS/ETS® 14.2 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS/ETS® 14.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/ETS® 14.2 User's Guide

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

November 2016

Chapter 51
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.

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, 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 3633 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.

The statements that follow enable you to access the prices for oil 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 51.1.

```
options validvarname=any;
title 'Retrieve Data for Oil India Limited Prices';
libname _all_ clear;

libname quan sasequan "%sysget(QUANDL)"
   OUTXML=oiltd
   XMLMAP="%sysget(QUANDL)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 51.1 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 filename, and the file extension (.xml) is excluded from the filename 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

```
C:\quandata\
```

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

```
C:\quandata\OIL_GSA.sas7bdat
```

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

```
libname quan sasequan "C:\quandata\"
   OUTXML=oiltd
   XMLMAP="C:\quandata\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 pathname that the XMLMAP= option specifies. The SASEQUAN engine appends a numeral to the XML filename 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 51.1. 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 GPLOT 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 51.1 summarizes the options that the SASEQUAN engine uses. In addition, there is one required option: API_KEY='*quan_api_key*'.

Table 51.1 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.
CONNECT=	Specifies whether or not you need 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, and 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, then the SASEQUAN engine appends the positional integer (‘1’ for 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, CUMUL, NORMALIZE, and NONE; the default is NONE. See Table 51.2 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 on 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.

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.

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 filename, as in the following, see the section "Getting Started: SASEQUAN Interface Engine" on page 3620:

```
FILENAME MyMap "C:\quandata\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 3631. The SET statement ("Getting Started: SASEQUAN Interface Engine" on page 3620) 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 filename 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 filename has the numeral 1 appended to the OUTXML filename. By default, OUTXML=QUAN, which creates a file named QUAN1.xml in the current working directory. The SAS data set 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.

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

COLLAPSE=DAILY | WEEKLY | MONTHLY | QUARTERLY | ANNUAL | NONE
specifies the frequency to which you want to collapse the reporting frequency.

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.

COLLAPSE= is optional. 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 to 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 (no longer supported) 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 using the Quandl Excel Add-In. SASEQUAN uses the QUANDL data sets API to request each time series in the IDLIST option, and enables 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 following URL: **`https://www.quandl.com/docs/api#data-organization`**.

NOTE: COLLAPSE=MONTHLY reports the daily, weekly, and monthly native frequencies of the time series at a monthly frequency (collapse frequency). If an annual native frequency time series

is specified 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 more than one Quandl code is specified in the IDLIST= option.

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

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 a secure connection, 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.

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.

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 an time series, enter the following URL in your web browser, and click on the Quandl code. The output gives you the list of available time series and their native frequencies and descriptions, including the "Frequency" field, which shows the native frequency of that time series.

```
https://www.quandl.com/resources/data-sources
```

You can use the data-sources URL to obtain the necessary information to request time series data for any of the listed data sources that you want to query.

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.

START='*quan_startdate*'
specifies the start date for the time series in the format `'YYYY-MM-DD'`. START= is optional, and the default is 1776-07-04 (earliest available). The date must be enclosed in single quotation marks.

END='*quan_enddate*'
specifies the end date for the time series in the format `'YYYY-MM-DD'`. END= is optional, and the default is 9999-12-31 (latest available). The date must be enclosed in single quotation marks.

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*
specifies the maximum number of rows (time series observations) to return, which is an integer between 1 and 100,000. ROWS= is optional, and the default is ROWS=100000.

SORT=ASC | DESC
specifies the order in which to sort the date of time series observations.

ASC	sorts time series observations in ascending date order.
DESC	sorts time series observations in descending date order.

SORT= is optional, and the default is SORT=ASC.

TRANS=CUMUL | DIFF | NORMALIZE | RDIFF | NONE
TRANSFORMATION=CUMUL | DIFF | NORMALIZE | RDIFF | NONE
specifies the data value transformation.

CUMUL	performs the cumulative function.
DIFF	performs the difference function.
NORMALIZE	performs the normalize function.
RDIFF	performs the ratio difference function.
NONE	does no transformation on the data.

TRANS= is optional, and the default is TRANS=NONE. The details of the arguments and the corresponding function formulas are presented in Table 51.2.

Table 51.2 Quandl Transformation Codes

Trans Code	Description	Formula
cumul	Cumulative	$x_t + x_{t-1} + \cdots + x_{t-N}$
diff	Difference	$x_t - x_{t-1}$
normalize	Normalize	$(\frac{x_t}{x_{t-N}}) \times 100$
rdiff	Ratio difference	$(\frac{x_t - x_{t-1}}{x_{t-1}})$

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 ($N = 52$).

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. The unique code for any data set is always visible on the data set page beside 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 Time Series Data

To obtain a list of the available sources of Quandl data, enter the following URL in your web browser. Table 51.3 shows some of the financial sources available (in the order shown on the Quandl website), which you can see by clicking on the "Financial Data" category.

```
https://www.quandl.com/resources/data-sources?auth_token=your_quan_apikey
```

Table 51.3 Some Available Sources of Financial Data

Quandl Code (Source)	Name
OFDP	See Open Financial Data Project
YAHOO	Yahoo Finance
GOOG	Google Finance
PSYCH	PsychSignal
NASDAQ	NASDAQ
NSE	National Stock Exchange of India
NYX	See NYSE Euronext
ICE	IntercontinentalExchange
SPDJ	S&P Dow Jones Indices
EUREX	Eurex
QUANDL	Quandl
LIFFE	See Euronext LIFFE
PXDATA	Price-Data.com
SHFE	Shanghai Futures Exchange
MGEX	Minneapolis Grain Exchange
CBOE	Chicago Board Options Exchange
SGX	Singapore Exchange
MX	Montréal Exchange
EUREKA	Eurekahedge
WFE	World Federation of Exchanges
ASX	Australian Securities Exchange
SIX	Swiss Exchange
WGC	World Gold Council
SANDP	See Standard & Poor's
WSJ	The Wall Street Journal
BITCOIN	Bitcoin Charts
MYX	Bursa Malaysia
OSE	See Osaka Securities Exchange
ABMI	Asian Bond Market Initiative
ODE	Osaka Dojima Commodity Exchange
AMMANSE	Amman Stock Exchange
BCHAIN	Block Chain
UCR	Unicorn Research Corporation
ML	Merrill Lynch
UIS	University of Stavanger
NIKKEI	Nikkei Group
WREN	Wren Research
MONEYTREE	MoneyTree Report
PUP	Policy Uncertainty Project
NASOMXNORDIC	NASDAQ OMX Nordic
PHILSE	Philippine Stock Exchange
THAISE	Stock Exchange of Thailand
BUDAPESTSE	Budapest Stock Exchange
RENCAP	Renaissance Capital

Useful Lists for Easy Downloading of Quandl Time Series Data

To get a list of Quandl codes for the available time series for a specific source, enter the following URL in your web browser and click on the category or the particular link for that source:

```
https://www.quandl.com/resources/useful-lists
```

Table 51.4 shows the list of Quandl codes for the 30 companies whose stocks make up the Dow Jones Industrial Average market index.

Table 51.4 Quandl Codes for Dow Jones Industrial Average Constituents

Ticker Symbol	Quandl Code	Name
AXP	GOOG/NYSE_AXP	American Express
BA	GOOG/NYSE_BA	Boeing
CAT	GOOG/NYSE_CAT	Caterpillar
CSCO	GOOG/NASDAQ_CSCO	Cisco Systems
CVX	GOOG/NYSE_CVX	Chevron
DD	GOOG/NYSE_DD	DuPont
DIS	GOOG/NYSE_DIS	Walt Disney
GE	GOOG/NYSE_GE	General Electric
GS	GOOG/NYSE_GS	Goldman Sachs
HD	GOOG/NYSE_HD	The Home Depot
IBM	GOOG/NYSE_IBM	IBM
INTC	GOOG/NASDAQ_INTC	Intel
JNJ	GOOG/NYSE_JNJ	Johnson & Johnson
JPM	GOOG/NYSE_JPM	JPMorgan Chase
KO	GOOG/NYSE_KO	Coca-Cola
MCD	GOOG/NYSE_MCD	McDonald's
MMM	GOOG/NYSE_MMM	3M
MRK	GOOG/NYSE_MRK	Merck
MSFT	GOOG/NASDAQ_MSFT	Microsoft
NKE	GOOG/NYSE_NKE	Nike
PFE	GOOG/NYSE_PFE	Pfizer
PG	GOOG/NYSE_PG	Procter & Gamble
T	GOOG/NASDAQ_T	AT&T
TRV	GOOG/NYSE_TRV	Travelers
UNH	GOOG/NYSE_UNH	UnitedHealth Group
UTX	GOOG/NYSE_UTX	United Technologies
V	GOOG/NYSE_V	Visa
VZ	GOOG/NYSE_VZ	Verizon
WMT	GOOG/NYSE_WMT	Wal-Mart
XOM	GOOG/NYSE_XOM	ExxonMobil

Available Time Series for Each Quandl Code

Quandl data sets can contain different numbers of time series at various frequencies. To get the list of time series for a specific source, use the Quandl search capability by entering the following URL in your web browser and specify the source ID that corresponds to that Quandl code. For example, the following URL retrieves a list of all time series for the United Nations Office on Drugs and Crime (UNODC, source_ids=692):

```
https://www.quandl.com/search/*?source_ids=692&api_key=your_quan_apikey
```

A shortcut to this source is at the URL **https://www.quandl.com/resources/data-sources**.

Under International Organizations, you can see the UNODC code, which links to the list of available time series and their native frequencies and descriptions. You can use the data-sources URL to obtain the necessary information to request time series data for any of the listed data source that you want to query.

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 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 filename, the second data set has a 2 concatenated to the filename, 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 3624.

SAS XML Map File

The XML map that (by default) is automatically created is assigned the full pathname 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 51.1: 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 51.1.1.

```
options validvarname=any;

title 'Historical Prices for Oil India Limited';
libname _all_ clear;
libname mylib "U:\quan940\doc\";

libname myQoil sasequan "%sysget(QUANDL)"
   apikey='XXXXXXXXXXXXXXXXXXX'
   idlist='NSE/OIL'
   format=XML
   outXml=oil
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(QUANDL)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 51.1.1 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 51.2: 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 51.2.1.

```
title 'Retrieve Data for Three Time Series: FRED/CP, FRED/GDP, FRED/TOTALSL';
libname _all_ clear;
options validvarname=any;
libname mylib "U:\quan940\doc\";

libname myQ3 sasequan "%sysget(QUANDL)"
   OUTXML=fred3
   AUTOMAP=replace
   MAPREF=MyMap
   XMLMAP="%sysget(QUANDl)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 51.2.1 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	1273.2	14384.1	2574.61
2	2009-12-31	1374.4	14566.5	2555.39
3	2010-03-31	1450.2	14681.1	2537.81
4	2010-06-30	1436.8	14888.6	2520.87
5	2010-09-30	1499.1	15057.7	2520.13
6	2010-12-31	1494.5	15230.2	2647.20
7	2011-03-31	1350.2	15238.4	2672.78
8	2011-06-30	1423.1	15460.9	2695.36
9	2011-09-30	1430.4	15587.1	2720.46
10	2011-12-31	1507.1	15785.3	2758.27
11	2012-03-31	1727.4	15973.9	2791.40
12	2012-06-30	1654.8	16121.9	2837.00
13	2012-09-30	1686.6	16227.9	2873.26
14	2012-12-31	1663.9	16297.3	2920.35
15	2013-03-31	1679.5	16475.4	2965.73
16	2013-06-30	1658.1	16541.4	3003.45
17	2013-09-30	1685.6	16749.3	3017.04

Example 51.3: Retrieving Data for the JASDAQ-TOP20 Exchange Traded Fund (ETF)

This example shows how to use one Quandl code, GOOG/TYO_1551, to retrieve the price and yield performance of the JASDAQ-TOP20 Exchange Traded Fund (ETF) in Japan, starting January 1, 2014, and ending March 26, 2014, with a daily native frequency. The output is shown in Output 51.3.1.

```
options validvarname=any;

title 'JASDAQ-TOP20 ETF, Five Time Series';
libname _all_ clear;
libname mylib "U:\quan940\doc\";

libname myTOP20 sasequan "%sysget(QUANDL)"
   apikey='XXXXXXXXXXXXXXXXXXXX'
   idlist='GOOG/TYO_1551'
   format=XML
   outXml=jasdaq
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(QUANDL)jasdaq.map"
   start='2014-01-01'
   end='2014-03-26'
   ;

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

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

Output 51.3.1 JASDAQ-TOP20 ETF

JASDAQ-TOP20 ETF, Five Time Series

Obs	date	Open	High	Low	Close	Volume
1	2014-01-06	6200	6260	6020	6080	19310
2	2014-01-07	6020	6040	5900	5910	11470
3	2014-01-08	6020	6150	5990	6140	11300
4	2014-01-09	6100	6190	6070	6130	13050
5	2014-01-10	6150	6190	6060	6100	13980
6	2014-01-14	5950	6080	5920	5950	21170
7	2014-01-15	6100	6100	6000	6020	6630
8	2014-01-16	6070	6100	5900	5940	8990
9	2014-01-17	5910	6010	5850	6000	8130
10	2014-01-20	6020	6030	5930	5950	4650
11	2014-01-21	5980	6110	5960	6060	14470
12	2014-01-22	6100	6130	6060	6110	7630
13	2014-01-23	6110	6130	5940	5960	11220
14	2014-01-24	5820	6050	5700	5920	23250
15	2014-01-27	5560	5710	5560	5580	18300
16	2014-01-28	5600	5790	5580	5600	11680
17	2014-01-29	5780	5920	5760	5880	12980
18	2014-01-30	5680	5770	5610	5700	8040
19	2014-01-31	5780	5820	5550	5620	8440
20	2014-02-03	5570	5580	5400	5430	14610
21	2014-02-04	5140	5490	5080	5300	40470
22	2014-02-05	5500	5550	5100	5290	18350
23	2014-02-06	5340	5690	5340	5570	11950
24	2014-02-07	5670	5700	5550	5590	8070
25	2014-02-10	5570	5720	5570	5710	8370
26	2014-02-12	5880	5900	5770	5830	12160
27	2014-02-13	5820	5820	5650	5710	5240
28	2014-02-14	5660	5700	5480	5520	6990
29	2014-02-17	5530	5600	5400	5480	3340
30	2014-02-18	5580	5810	5580	5810	8150
31	2014-02-19	5720	5880	5690	5690	5060
32	2014-02-20	5700	5910	5610	5640	15270
33	2014-02-21	5700	5770	5640	5640	8550
34	2014-02-24	5660	5880	5650	5810	9100
35	2014-02-25	5900	6200	5850	6080	37800

Example 51.4: Collapsing Data for the JASDAQ-TOP20 Exchange Traded Fund (ETF)

This example shows how to collapse the daily data to a weekly interval by using the same Quandl code as in Example 51.3, GOOG/TYO_1551, to retrieve the price and yield performance of the JASDAQ-TOP20 Exchange Traded Fund (ETF) in Japan, starting January 1, 2014, and ending March 26, 2014, with a daily native frequency, collapsing to a weekly frequency by using the COLLAPSE= option. The output is shown in Output 51.4.1.

```
options validvarname=any;

title 'JASDAQ-TOP20 ETF, COLLAPSE=WEEKLY Option';
libname _all_ clear;
libname mylib "U:\quan940\doc\";

libname myTOP20 sasequan "%sysget(QUANDL)"
   apikey='XXXXXXXXXXXXXXXXXXXX'
   idlist='GOOG/TYO_1551'
   format=XML
   outXml=jasdaqW
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(QUANDL)jasdaqw.map"
   start='2014-01-01'
   end='2014-03-26'
   collapse=weekly
   ;

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

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

Output 51.4.1 JASDAQ-TOP20 ETF, with COLLAPSE=WEEKLY

JASDAQ-TOP20 ETF, COLLAPSE=WEEKLY Option

Obs	date	Open	High	Low	Close	Volume
1	2014-01-12	6150	6190	6060	6100	13980
2	2014-01-19	5910	6010	5850	6000	8130
3	2014-01-26	5820	6050	5700	5920	23250
4	2014-02-02	5780	5820	5550	5620	8440
5	2014-02-09	5670	5700	5550	5590	8070
6	2014-02-16	5660	5700	5480	5520	6990
7	2014-02-23	5700	5770	5640	5640	8550
8	2014-03-02	5830	5830	5650	5670	4420
9	2014-03-09	5870	5890	5850	5850	3740
10	2014-03-16	5550	5620	5420	5470	9030
11	2014-03-23	5300	5340	5130	5140	4780
12	2014-03-30	4860	5010	4850	4950	9100

Example 51.5: Transforming Data for the JASDAQ-TOP20 Exchange Traded Fund (ETF)

This example shows how to transform the daily data by using the diff transformation and the same Quandl code as in Example 51.3 and Example 51.4, GOOG/TYO_1551, to retrieve the price and yield performance of the JASDAQ-TOP20 Exchange Traded Fund (ETF) in Japan. Specify a range by using START='2014-01-01' and END='2014-03-26', a transformation function by using TRANS=DIFF, and a collapse frequency by using COLLAPSE=WEEKLY. The output is shown on Output 51.5.1.

```
options validvarname=any;

title 'JASDAQ-TOP20 ETF, TRANS=DIFF Option';
libname _all_ clear;
libname mylib "U:\quan940\doc\";

libname myTOP20 sasequan "%sysget(QUANDL)"
   apikey='XXXXXXXXXXXXXXXXXXXX'
   idlist='GOOG/TYO_1551'
   format=XML
   outXml=jasdaqX
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(QUANDL)jasdaqX.map"
   start='2014-01-01'
   end='2014-03-26'
   collapse=weekly
   trans=diff
   ;

data mylib.jasdaqX;
   set myTOP20.jasdaqX;
```

```
run;

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

Output 51.5.1 JASDAQ-TOP20 ETF, Weekly Data with TRANS=DIFF

JASDAQ-TOP20 ETF, TRANS=DIFF Option

Obs	date	Open	High	Low	Close	Volume
1	2014-01-19	-240	-180	-210	-100	-5850
2	2014-01-26	-90	40	-150	-80	15120
3	2014-02-02	-40	-230	-150	-300	-14810
4	2014-02-09	-110	-120	0	-30	-370
5	2014-02-16	-10	0	-70	-70	-1080
6	2014-02-23	40	70	160	120	1560
7	2014-03-02	130	60	10	30	-4130
8	2014-03-09	40	60	200	180	-680
9	2014-03-16	-320	-270	-430	-380	5290
10	2014-03-23	-250	-280	-290	-330	-4250
11	2014-03-30	-440	-330	-280	-190	4320

Example 51.6: Reading 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 DOE/RWTC, BUNDESBANK/BBK01_WT5511, and YAHOO/INDEX_GSPC to retrieve oil, gold, and stock prices. There are eight time series (one for oil, one for gold, and six for S&P500), taken from three different Quandl data sets: DOE/RWTC, BUNDESBANK/BBK01_WT5511, and YAHOO/INDEX_GSPC, respectively. Because the Oil, Gold, and S&P500 columns are all from daily native frequency data sets, you can use the collapse frequency "Annual" to minimize the missing values in the output. In the following example, specify a range by using START='1968-12-31' and END='2014-03-27', and specify a collapse frequency by using COLLAPSE=ANNUAL. The output is shown in Output 51.6.1.

```
options validvarname=any;

title 'Oil, Gold and S&P Index Stock Time Series Using the COLLAPSE= Option';
libname _all_ clear;
libname mylib "U:\quan940\doc\";

libname mysup sasequan "%sysget(QUANDL)"
   apikey='XXXXXXXXXXXXXXXXXXX'
   idlist='DOE/RWTC,BUNDESBANK/BBK01_WT5511,YAHOO/INDEX_GSPC'
   format=XML
   outXml=Tsupe
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(QUANDL)Tsupe.map"
   start='1968-12-31'
   end='2014-03-27'
   collapse=annual
   ;

data mylib.Tsupe;
   set mysup.Tsupe;
   label Value_1 = "WTI Crude Oil Spot Price Cushing, OK FOB";
   label Value_2= "Gold Price (USD)";
   label Open_3= "S&P 500 Index Open";
   label High_3="S&P 500 Index High";
   label Low_3= "S&P 500 Index Low";
   label Close_3="S&P 500 Index Close";
   label Volume_3="S&P 500 Index Volume";
   label 'Adjusted Close_3'n ="S&P 500 Index Adjusted Close";
run;

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

Output 51.6.1 Reading from Multiple Quandl Data Sets: Oil, Gold, and Stock Prices Using COLLAPSE= Option

Oil, Gold and S&P Index Stock Time Series Using the COLLAPSE= Option

Obs	date	WTI Crude Oil Spot Price Cushing, OK FOB	Gold Price (USD)	S&P 500 Index Open	S&P 500 Index High	S&P 500 Index Low	S&P 500 Index Close	S&P 500 Index Volume	S&P 500 Index Adjusted Close
1	1968-12-31	.	41.950	103.80	104.61	102.98	103.86	13130000	103.86
2	1969-12-31	.	35.210	91.60	92.94	91.15	92.06	19380000	92.06
3	1970-12-31	.	37.375	92.27	92.79	91.36	92.15	13390000	92.15
4	1971-12-31	.	43.640	102.09	102.09	102.09	102.09	14040000	102.09
5	1972-12-31	.	64.700	116.93	118.77	116.70	118.05	27550000	118.05
6	1973-12-31	.	112.250	97.54	98.30	95.95	97.55	23470000	97.55
7	1974-12-31	.	187.500	67.16	69.04	67.15	68.56	20970000	68.56
8	1975-12-31	.	140.250	89.77	90.75	89.17	90.19	16970000	90.19
9	1976-12-31	.	134.550	106.88	107.82	106.55	107.46	19170000	107.46
10	1977-12-31	.	165.600	94.94	95.67	94.44	95.10	23560000	95.10
11	1978-12-31	.	224.500	96.28	97.03	95.48	96.11	30030000	96.11
12	1979-12-31	.	524.000	107.84	108.53	107.26	107.94	31530000	107.94
13	1980-12-31	.	589.500	135.33	136.76	134.29	135.76	41210000	135.76
14	1981-12-31	.	400.000	122.30	123.42	121.57	122.55	40780000	122.55
15	1982-12-31	.	448.000	140.34	140.78	140.27	140.64	42110000	140.64
16	1983-12-31	.	381.500	164.86	165.05	164.58	164.93	71840000	164.93
17	1984-12-31	.	309.000	166.26	167.34	166.06	167.24	80260000	167.24
18	1985-12-31	.	327.000	210.68	211.61	210.68	211.28	112700000	211.28
19	1986-12-31	17.93	390.900	243.37	244.03	241.28	242.17	139200000	242.17
20	1987-12-31	16.74	486.500	247.84	247.86	245.22	247.08	170140000	247.08
21	1988-12-31	17.12	410.150	279.39	279.78	277.72	277.72	127210000	277.72
22	1989-12-31	21.84	401.000	350.68	353.41	350.67	353.40	145940000	353.40
23	1990-12-31	28.48	391.000	328.71	330.23	327.50	330.22	114130000	330.22
24	1991-12-31	19.15	353.400	415.14	418.32	412.73	417.09	247080000	417.09
25	1992-12-31	19.49	332.900	438.82	439.59	435.71	435.71	165910000	435.71
26	1993-12-31	14.19	390.650	468.66	470.75	466.45	466.45	168590000	466.45
27	1994-12-31	17.77	382.500	461.17	462.12	459.24	459.27	256260000	459.27
28	1995-12-31	19.54	386.700	614.12	615.93	612.36	615.93	321250000	615.93
29	1996-12-31	25.90	369.550	753.85	753.95	740.74	740.74	399760000	740.74
30	1997-12-31	17.65	289.200	970.84	975.02	967.41	970.43	467280000	970.43
31	1998-12-31	12.14	287.450	1231.93	1237.18	1224.96	1229.23	719200000	1229.23
32	1999-12-31	25.76	290.850	1464.47	1472.42	1458.19	1469.25	374050000	1469.25
33	2000-12-31	26.72	272.650	1334.22	1340.10	1317.51	1320.28	1035500000	1320.28
34	2001-12-31	19.96	276.500	1161.02	1161.16	1148.04	1148.08	943600000	1148.08
35	2002-12-31	31.21	342.750	879.39	881.93	869.45	879.82	1088500000	879.82

References

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

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

Kamel, T. (2014b). "Useful Lists." Accessed March 14, 2014. https://www.quandl.com/resources/useful-lists/.

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

Subject Index

Syntax Index