THE POWER TO KNOW.

SAS/ETS® 14.2 User's Guide
The SASERAIN 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 52
The SASERAIN Interface Engine

Contents

Overview: SASERAIN Interface Engine

The SASERAIN interface engine enables SAS users to retrieve weather data from the World Weather Online website. This website offers access to time series of weather data such as temperature, precipitation (rainfall), weather description, weather icon, and wind speed. These time series are updated at intervals that the user selects. The weather time series on the World Weather Online website contain observation or measurement periods that are associated with data values.

The SASERAIN interface engine uses the LIBNAME statement to enable you to download World Weather Online data and to specify which weather data time series you want to retrieve based on location. You can then use the SAS DATA step to perform further subsetting and to store the resulting time series in a SAS data set.

There are two types of major weather application interfaces (APIs) that return World Weather Online data for the SASERAIN engine. The first type is a local weather API that returns forecasting data and current conditions data, which usually start with today and end with tomorrow's forecast. You can request up to 5 days of free (nonpremium) local weather forecast data or 15 days of premium local weather forecast data. The

SASERAIN engine for SAS/ETS 14.2 supports both the premium and nonpremium local weather APIs, and the default range for the SASERAIN engine is 2 days. You can use the premium local weather forecast API if you have subscribed to the premium service and have also specified your premium API key. The premium API key provides a longer maximum date range of 15 days than the 5-day maximum for the nonpremium API key.

The second type of API is a historical weather API that returns past weather. When you have a premium subscription, you can use a range that starts as early as July 1, 2008. When you use the nonpremium past weather API (the default), your start date must be within the past 60 days. For nonpremium past weather, when you specify a range that is not within the past 60 days, no data are returned, and a fatal error occurs.

When no dates are specified, the default type of data that the SASERAIN interface engine returns is the local forecast weather data. **NOTE:** The SASERAIN interface uses the past weather API whenever a range of dates is specified by a start date and an end date.

You can choose to retrieve the following types of data for a single location or multiple locations:

- current conditions only
- local weather forecast only
- both current conditions and the local weather forecast
- 24-hour weather forecast only (the frequency is auto-set to 3 hours over one 24-hour period)
- historical (past) weather for a specified date range

The SASERAIN interface engine supports Linux X64 (LAX) and Windows. Although the SASERAIN engine uses the World Weather Online API, it is not endorsed or certified by World Weather Online. By using the SASERAIN interface engine, you are agreeing to comply with the World Weather Online terms of use, which are described on the web page at the following URL:

https://www.worldweatheronline.com/terms-and-conditions.aspx

Getting Started: SASERAIN Interface Engine

You can query the World Weather Online database to retrieve the observations or data values for a list of time series by specifying the World Weather Online code for the location (q-code). The World Weather Online q-code consists of a location code such as one for City and Country, latitude and longitude, IP address, US zip code, UK/Canadian postal code, or airport code (IATA). To specify more than one location, list each q-code in the QUERY= option, and separate the locations with a semicolon. Neither a comma nor a blank can be used as a separator between the q-codes, because one q-code can contain any number of commas or blanks.

You must also specify your unique World Weather Online API key (authentication token). To obtain your own free unique API key, visit the World Weather Online website at the following URL:

https://developer.worldweatheronline.com/login.aspx

For more information about the web service (including pricing and premium service information), visit the website at the following URL:

https://developer.worldweatheronline.com/api/faq.aspx

The World Weather Online API key is a 29-character mixed-case alphanumeric string, such as "abCDefghijklmnopqrst123456789," and is represented by 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX' in the APIKEY= option in the following example. In addition, the example URLs in this section and in the section "Examples: SASERAIN Interface Engine" on page 3659 use the same World Weather Online API key as the argument *your_rain_apikey*.

After you have your assigned World Weather Online API key and have agreed to the World Weather Online terms of use, you can use your API key to access the World Weather Online data, as shown in the following example.

The statements that follow enable you to access the weather for London, Paris, and Dubai. For brevity of output, the request is for only one day (NUM_OF_DAYS=1), which starts with today. The FX24=YES option returns observations at a frequency of every 3 hours with an additional observation for the 24-hour average (the value of the TIME variable is 24), and the observations are sorted in chronological order. For brevity, only the current conditions output is shown in Figure 52.1.

```
options validvarname=any;
title 'Retrieve Weather Data for London, Paris, and Dubai';
libname _all_ clear;

libname mylib "U:\rain940\doc\";

libname rain saserain "%sysget(RAIN_DATA)"
   QUERY='London,United Kingdom;Paris,France;Dubai,United Arab Emirates'
   FX24=yes
   CONDITIONS=yes
   OUTXML=tricky
   AUTOMAP=replace
   MAPREF=MyMap
   XMLMAP="%sysget(RAIN_DATA)tricky.map"
   APIKEY='XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
   NUM_OF_DAYS=1
   FORMAT=xml;

data mylib.my24a;
   set rain.tricky;
run;
proc contents data=mylib.my24a; run;
proc print data=mylib.my24a; run;

libname condo "U:\rain940\test\";

data mylib.mycca;
   set condo.cc_tricky;
run;
```

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

Figure 52.1 Current Conditions for London, Paris, and Dubai

Retrieve Weather Data for London, Paris, and Dubai

Obs	AreaName	Country	Region	weatherDesc	winddir16Point	observation_time	oc
1	London	United Kingdom	City of London, Greater London	Clear	WSW	19:19:00	1
2	Paris	France	Ile-de-France	Clear	N	19:19:00	2
3	Dubai	United Arab Emirates	Dubai	Clear	ENE	19:19:00	3

Obs	latitude	longitude	temp_C	temp_F	weatherCode	windspeedMiles	windspeedKmph
1	51.517	-0.106	22	72	113	15	24
2	48.867	2.333	21	70	113	6	9
3	25.252	55.280	34	93	113	11	17

Obs	winddirDegree	precipMM	humidity	visibility	pressure	cloudcover	FeelsLikeC	FeelsLikeF
1	250	0	43	10	1013	0	24	76
2	360	0	53	10	1017	0	21	70
3	70	0	53	10	1004	0	40	104

The XML data that the World Weather Online website returns are placed in a file that is named by the OUTXML= option—in this case, *TRICKY1.xml*. **NOTE:** The SASERAIN engine appends a numeral to the XML filename, and the file extension (.xml) is excluded from the filename that appears in the OUTXML= option. When the SET statement is executed, the XML data are read into a SAS data set named TRICKY.sas7bdat, which resides in the location given inside the string enclosed in double quotation marks in the SASERAIN LIBNAME statement. So, in the preceding example, if the RAIN_DATA environment variable is set to **U:\rain940\test**, then the SAS data set is named **U:\rain940\test\TRICKY.sas7bdat**. An equivalent LIBNAME statement that does not use any environment variables could be as follows:

```
libname rain saserain "U:\rain940\test\"
   QUERY='London,United Kingdom;Paris,France;Dubai,United Arab Emirates'
   NUM_OF_DAYS=1
   FX24=yes
   CONDITIONS=yes
   OUTXML=tricky
   AUTOMAP=replace
   MAPREF=MyMap
   XMLMAP="U:\rain940\test\tricky.map"
   APIKEY='XXXXXXXXXXXXXXXXXXX'
   FORMAT=xml;
```

You could also use either a SAS macro variable or a system environment variable to store the value of your World Weather Online 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 SASERAIN engine appends a numeral to the XML filename to indicate the position of the World Weather Online location code in the QUERY= option.

The QUERY= option specifies the list of World Weather Online locations that you want to retrieve weather data for. This option accepts a string, enclosed in single quotation marks, that denotes a list of one or more World Weather Online locations that you select (keep) in the resulting SAS data set. The result, TRICKY, is named in the DATA step and is shown in Figure 52.1. The preceding example uses three World Weather Online location codes. London, which is in the first position of the QUERY= option, has the numeral 1 appended to the name of the XML file, resulting in *TRICKY1.xml*. Paris is in the second position of the QUERY= option, so the numeral 2 is appended to the name of the XML file, resulting in *TRICKY2.xml*. Dubai is in the third position of the QUERY= option, so the numeral 3 is appended to the name of the XML file, resulting in *TRICKY3.xml*. The SASERAIN engine merges the three XML files to produce one merged output data set named TRICKY.sas7bdat. The current conditions data set is named CC_TRICKY. The second DATA step uses the SET statement to read the current conditions data into a new data set named MYCCA. These data are shown in Figure 52.1.

It is more efficient to use the DATA step to store your World Weather Online 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 SASERAIN libref directly, as in the statement

```
proc print data=rain.tricky;
```

This statement uses the member name, TRICKY, in the PROC PRINT statement which invokes the RAIN libref to run the SASERAIN engine. This usage of the member name, TRICKY, corresponds to specifying the OUTXML=TRICKY option. Although using this statement might seem easier, it is not as efficient, because every time you use the SASERAIN libref, the SASERAIN 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. For another example that uses more SASERAIN LIBNAME statement options, see the section "Examples: SASERAIN Interface Engine" on page 3659.

Syntax: SASERAIN Interface Engine

The SASERAIN interface engine uses standard engine syntax to read the observations or data values for one or more World Weather Online data sets that can each contain one or more time series. Table 52.1 summarizes the options that the SASERAIN engine uses. In addition, there is one required option: APIKEY='*rain_api_key*'. If your API key is a premium key, then specify the PREMIUM=YES option. When you use the PREMIUM=YES option with the APIKEY= option that specifies a premium API key, you gain access to a longer date range of data availability and more frequency options (the monthly averaged climate data are added).

Table 52.1 Summary of LIBNAME *libref* SASERAIN Options

Option	Description
APIKEY=	Specifies the required World Weather Online access key that enables you to access the data that the World Weather Online website provides
AUTOMAP=	Specifies whether or not to overwrite the existing XML map file
CONDITIONS=	Specifies whether or not to return only the current weather conditions upon output. CONDITIONS=YES means that variables for both the current conditions and the weather forecast appear in the output. The default (NO) means that only the local weather forecast variables appear in the output.
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.
DATE=	Specifies the beginning date for past weather data for the specified range: specify the start date in `'YYYY-MM-DD'` format. The range must be less than 60 days prior to today's date for nonpremium users, and must be on or after July 1, 2008, for premium users.
DAY=	Specifies that the local weather forecast is to be current weather, not past weather. When you specify either today or tomorrow, you get today's weather forecast. This is used with the NUM_OF_DAYS= option to specify a range for obtaining local weather forecast data.
DEBUG=	Specifies whether or not to include diagnostic message logging in the SAS log window
ENDDATE=	Specifies the end date for past weather for the specified range: specify the end date in `'YYYY-MM-DD'` format. The end date must be within the range of 60 days prior to today's date.
FORECAST=	Specifies whether or not to return the weather forecast for a given postal code, zip code, and latitude/longitude values
FORMAT=	Specifies a file extension that indicates the type of file to retrieve. Only XML is supported for the SASERAIN interface engine.
FREQ=	Specifies the frequency (interval) of the selected weather forecast data as a character string, such as DAILY, 24HOURLY, HOURLY, 3HOURLY, 6HOURLY, 12HOURLY, or DAY/NIGHT
FX24=	Specifies whether or not to return the 24-hour weather forecast at a three-hour interval for a given location (city and country, postal code, zip code, or latitude and longitude)
NUM_OF_DAYS=	Specifies the number of days to report (starting from today). This is used for reading the local weather forecast data. The default for the SASERAIN engine is set to 2 days, and the maximum is 5 days (nonpremium weather API) or 15 days (premium weather API) of forecast data.

Table 52.1 *continued*

Option	Description
OUTCC=	Specifies the name of the current conditions SAS data set, which contains current conditions data returned by the World Weather Online API. This option is ignored when CONDITIONS=NO. For more information, see the CONDITIONS= option.
OUTXML=	Specifies the name of the SAS data set and the XML file, which usually contains the weather forecast data returned by the World Weather Online API. When you do not specify the OUTCC= option, the SASERAIN interface prepends 'CC_' to the name specified in the OUTXML= option to create the name for the current conditions SAS data set. See the OUTCC= option.
PREMIUM=	Specifies whether or not to use the premium subscription API from World Weather Online. Your API key needs to be a premium key if you specify YES. The default (NO) means that the output will request data from the World Weather Online nonpremium API.
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.
QUERY=	Specifies a required list of World Weather Online location codes. To select more than one location, list the World Weather Online query codes (q-codes), separated by semicolons. There is a limit of nine World Weather Online location codes in the QUERY= option. This is a required option.
TP=	Specifies the time period (interval) of the selected weather forecast data in number of hours: 1, 3, 6 (default), 12, or 24 hours.
XMLMAP=	Specifies the fully qualified filename for the XML map that the SASERAIN engine creates. This filename is usually the same as the one in the OUTXML= option.

The LIBNAME *libref* SASERAIN Statement

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

The LIBNAME statement assigns a SAS library reference (libref) to the physical path of the directory of World Weather Online data files in which the downloaded World Weather Online XML data are stored. The required *physical-name* argument specifies the location of the folder where your World Weather Online XML data reside. 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* SASERAIN statement.

APIKEY='*rain_apikey***'**
specifies the World Weather Online authentication token or access key that enables you to access the data that the World Weather Online website provides. This access key is a 29-character mixed-case alphanumeric string, and it is required. It must be enclosed in single quotation marks. You can request your *rain_apikey* by visiting the website at the following URL:

https://developer.worldweatheronline.com/auth/register

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 SASERAIN 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.

CONDITIONS=ONLYCC | YES | NO
specifies whether or not to return only current conditions data. CONDITIONS=ONLYCC enables the SASERAIN interface to output the current conditions data but not the forecast data. For more about current conditions, see Table 52.2.

ONLYCC specifies that only the current conditions be output.

YES specifies that the current conditions be output.

NO specifies that the current conditions variables be excluded from the output.

By default, the SASERAIN engine uses CONDITIONS=NO and FORECAST=YES. Specify CONDITIONS=YES to create both the current conditions output data set (named in the OUTCC= option) and the weather forecast output data set (named in the OUTXML= option). When the OUTCC= option is not specified, the prefix 'CC_' is added to the name specified in the OUTXML= option. For more information, see the FORECAST= and OUTCC= options. The SASERAIN engine issues a warning when both past weather and current conditions are selected in the same SASERAIN LIBNAME statement.

Table 52.2 Current Conditions Forecast Variables

Variable Name	Description
observation_time	Time in UTC `'hhmm tt'` format. For example: 06:45 AM or 11:34 PM.
temp_C	Temperature in degrees Celsius
windspeedMiles	Wind speed in miles per hour
windspeedKmph	Wind speed in kilometers per hour
winddirDegree	Wind direction in degrees
winddir16Point	Wind direction on a 16-point compass
weatherCode	Weather condition code
weatherDesc	Weather condition description
weatherIconUrl	URL for weather icon
precipMM	Precipitation in millimeters
precipInches	Precipitation in inches
humidity	Humidity in percentage
visibility	Visibility in kilometers
visibilityMiles	Visibility in miles
pressure	Atmospheric pressure in millibars
pressureInches	Atmospheric pressure in inches
cloudcover	Cloud cover in percentage

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.

DATE=*rain_date_start*

specifies the start date for requesting past (historical) weather data: specify `'YYYY-MM-DD'` (format for the *rain_date_start*). The start date must fall within the last 60 days from today for nonpremium use. The earliest start date for premium users is July 1, 2008.

DAY=TODAY | TOMORROW

specifies the start date for the local current weather forecast: specify today or tomorrow, but results are the same—they start today. If you want a start date other than today, then use the DATE= option. Use the NUM_OF_DAYS= option to specify the number of days to report.

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.

ENDDATE=*rain_date_enddate*

specifies the end date for the range to report past weather: `'YYYY-MM-DD'` (format for the *rain_date_enddate*). The end date must fall within the last 60 days when the SASERAIN engine uses the nonpremium past (historical) weather API. The earliest start date (DATE= option) for premium past weather is July 1, 2008, but the ENDDATE= option must have the same month and year as the start

date. The date must be enclosed in single quotation marks. The ENDDATE= option is not required, and the default range is two days.

FORECAST=YES | NO

specifies whether or not to return the weather forecast for a given location (city and country, postal code, zip code, or latitude and longitude values). By default, the SASERAIN engine uses FORECAST=YES. For more about weather forecast variables, see Table 52.3. When the type of data is not specified in the LIBNAME statement options, the SASERAIN engine defaults to normal weather forecast data and automatically defaults to the FX=YES option. Use either the FX24= option or the FX= option (but not both). When you specify FX24=YES, you do not need to specify any interval (FREQ= option) or any range specification, because the default is 24 hours of data at an interval of every 3 hours (and an extra observation for the 24-hour average).

Table 52.3 Weather Forecast Variables

Variable Name	Description
date	Local forecast date in `'YYYY-MM-DD'` format. For example: 2013-05-31.
maxtempC	Maximum temperature of the day in degrees Celsius
maxtempF	Maximum temperature of the day in degrees Fahrenheit
mintempC	Minimum temperature of the day in degrees Celsius
mintempF	Minimum temperature of the day in degrees Fahrenheit
uvIndex	Ultraviolet radiation index
time	Local time in `'hmm'` format. For example: 100 or 1500.
tempC	Temperature in degrees Celsius
tempF	Temperature in degrees Fahrenheit
windspeedMiles	Wind speed in miles per hour
windspeedKmph	Wind speed in kilometers per hour
windspeedKnots	Wind speed in knots
windspeedMeterSec	Wind speed in meters per second
winddirDegree	Wind direction in degrees
winddir16Point	Wind direction on a 16-point compass
weatherCode	Weather condition code
weatherDesc	Weather condition description
weatherIconUrl	URL for weather icon
precipMM	Precipitation in millimeters
precipinches	Precipitation in inches
humidity	Humidity in percentage
visibility	Visibility in kilometers
visibilityMiles	Visibility in miles
pressure	Atmospheric pressure in millibars
pressureInches	Atmospheric pressure in inches
cloudcover	Cloud cover in percentage
chanceofrain	Chance of rain (precipitation) in percentage
chanceofwindy	Chance of being windy in percentage
chanceofovercast	Chance of being cloudy in percentage
chanceofsunny	Chance of being sunny in percentage
chanceoffrost	Chance of frost in percentage

Table 52.3 *continued*

Variable Name	Description
chanceoffog	Chance of fog in percentage
chanceofsnow	Chance of snow in percentage
chanceofthunder	Chance of thunder in percentage

FORMAT=XML
specifies the format of the file to be retrieved from the World Weather Online website. Although World Weather Online can report data in many formats, the SASERAIN engine supports only the XML format.

FREQ=DAILY | HOURLY | 3HOURLY | 6HOURLY | 12HOURLY | 24HOURLY | DAY/NIGHT
specifies the frequency of the weather data. In World Weather Online weather forecast data, the highest frequency is hourly, and the lowest frequency is daily.

The FREQ= option is not required, and the default interval value is 6 hours.

FX24=YES | NO
specifies whether or not to return the 24-hour weather forecast at a three-hour interval for city/country, postal code, zip code, and latitude/longitude values. By default, the SASERAIN engine uses FX24=NO. When the type of data is not specified in the LIBNAME statement options, the SASERAIN engine defaults to normal weather forecast data and automatically defaults to the FX=YES option. **NOTE:** Use either the FX24= option or the FX= option (but not both). When you specify FX24=YES, you do not need to specify any interval (FREQ= option) or any range specification, because the default is 24 hours of data at an interval of 3 hours, but there is also an extra observation for the 24-hour averages for the reported variables.

MAPREF=*rain_xmlmapref*
specifies the fileref to use for the map assignment. For an example of the SASERAIN engine that uses the MAPREF= and XMLMAP= options in the FILENAME statement in order to assign a filename, as in the following statement, see the section "Examples: SASERAIN Interface Engine" on page 3659:

```
FILENAME MyMap "U:\rain940\test\gstart.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 folder. The SAS data set that is created (when the XML data are read into SAS) is placed in the folder specified by *physical-name,* and you can reference it by using the myLib libref in your SASERAIN LIBNAME statement. This is shown in the section "Examples: SASERAIN Interface Engine" on page 3659, inside the DATA step in the SET statement. The SET statement reads observations from the input data set myLib.GSTART and stores them in a SAS data set named HowCool.

NUM_OF_DAYS=*rain_numdays*
specifies the number of days to report local weather (starting from today). The maximum is five days, and the default is two days, unless you specify PREMIUM=YES. See the PREMIUM= option.

OUTCC=*rain_outcc*

specifies the name of the SAS data set where the current conditions data that are returned from the World Weather Online website are stored. When OUTCC= option is not specified, the SASERAIN interface stores the current conditions data in a SAS data set named by adding the prefix 'CC_' to the name specified in the OUTXML= option. If there is no request for current conditions data, then the OUTCC= option is ignored.

OUTXML=*rain_xmlfile*

specifies the name of both the XML file (downloaded) and the SAS data set created when the XML data are read into SAS. Each World Weather Online location code that is listed in the QUERY= option is given a positional numeral: 1 for the first code in the QUERY= option, 2 for the second code, and so on. The SASERAIN 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 World Weather Online location code is specified in the QUERY= option, the filename has the numeral 1 appended to the OUTXML filename. By default, OUTXML=RAIN, which creates a file named *RAIN1.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 SASERAIN statement.

PREMIUM=YES | NO

specifies whether or not to use the premium subscription API from World Weather Online. When you specify PREMIUM=YES, your apikey needs to be a premium key. The default (PREMIUM=NO) means that the output will request data from the nonpremium API from World Weather Online, but when you specify PREMIUM=YES, you can use the premium local weather forecast API if you have subscribed to the premium service and have also specified your premium API key in the APIKEY= option. The premium API key provides a longer maximum forecast date range of 15 days, compared to the maximum 5 days allowed for the nonpremium API key, and it provides a longer date range of available past weather data. See also the ENDDATE= option.

PROXY="*rain_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 *rain_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.

QUERY=*'rain_qcode_list'*

specifies the list of World Weather Online locations for the data sets that contain the time series to be included in the output SAS data set. There is a limit of nine World Weather Online location codes in the QUERY= option. The argument *'rain_qcode_list'* is semicolon-delimited and must be enclosed in single quotation marks. For example:

```
QUERY='QCODE1;QCODE2;...QcodeN'}
```

Each **QCODE** specifies a weather data location in one of the following location formats:

Latitude,Longitude	specifies the location of the selected weather forecast in decimal degrees (XX.XXX,XX.XXX).
UScityName,State	specifies the location of the selected US city and state.

cityName,Country	specifies the location of the selected city in the specified country, or if the location is in the United States, you can specify *cityName,State*.
IPaddress	specifies the location by using the Internet Protocol address in XXX.XXX.XXX.XXX format.
USzipcode	specifies the location by using the US zip code format.
UK_CANpostalcode	specifies the location by using the United Kingdom or Canadian postal code format.

You can specify a maximum of nine q-code locations in the QUERY= option, separated by semicolons. Each q-code can contain commas, blanks, or both. The QUERY= option is required.

TP=1 | 3 | 6 | 12 | 24
specifies the number of hours in a time period. In World Weather Online weather forecast data, the highest frequency is 1 (hourly), and the lowest frequency is 24 (daily).

The TP= option is not required, and the default interval value is 6 hours.

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

Details: SASERAIN Interface Engine

The SASERAIN interface engine enables SAS users to access time series data that are stored in World Weather Online data sets that the World Weather Online website provides. Every World Weather Online data set is identified by a unique location code ID (which you specify in the QUERY= option). For example, London (England) is uniquely identified by the latitude and longitude that are obtained by using the search API at the following URL:

```
http://api.worldweatheronline.com/free/v2/search.ashx?query=LONDON,
UNITED%20KINGDOM&key=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

When you specify the QUERY= option (for one to nine locations), the SASERAIN engine automatically calls the search API to find the unique latitude and longitude for each location that you want. If the request is ambiguous (too vague), then the SASERAIN engine issues a warning that it is using the best first match, and then lists the 10 possible matches that were searched. If the wrong latitude and longitude for a location were selected, you can rerun the SASERAIN engine with a different QUERY= option from the list of possibilities that best match your desired location. **NOTE:** It is best to specify latitude and longitude if you are having difficulty pinpointing your desired location.

World Weather Online API Key

The API key that is used in these examples, 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX', is for demonstration only. To successfully download data from the World Weather Online website, use your own

World Weather Online API key, which is a 29-character mixed-case alphanumeric string. You can request your own API key by visiting the website at the following URL:

https://developer.worldweatheronline.com/auth/register.

SAS Output Data Set

You can use a SAS DATA step to write the selected World Weather Online 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 SAS 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 name of each location whose weather data is read from the World Weather Online website.

The SASERAIN interface engine maintains the sort order, so the locations (q-codes) are sorted in the resulting SAS data set by the order that you specify in the QUERY= 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 SASERAIN libref to create a custom view of your data.

SAS OUTXML File

The SAS XML (XML format) data that are returned from the World Weather Online website are placed in a file that is named by the OUTXML= option. The SASERAIN interface engine creates a separate XML file for each World Weather Online code that you list in the QUERY= option. The engine numbers each data set's XML file in the order in which it appears in the QUERY= 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. When the QUERY= option contains more than one World Weather Online 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 are placed in the current working directory. The SAS data set created when the XML data are read into SAS is placed in the location specified by the *physical-name* in the LIBNAME *libref* SASERAIN statement, which is described in the section "The LIBNAME *libref* SASERAIN Statement" on page 3651.

SAS XML Map File

The XML map that (by default) is automatically created is assigned the full pathname that you specify in the XMLMAP= option in your LIBNAME *libref* SASERAIN 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 SASERAIN interface engine invokes the XMLV2 engine to create the map and to read the data into SAS.

Examples: SASERAIN Interface Engine

Example 52.1: Retrieving Weather Forecast Data for One Location

When you are specifying one location by city, it is important to also specify the country. Because spaces are allowed in city names and country names, a comma (without spaces) is required to separate the city name from the country name. The following statements enable you to access the World Weather Online data for Paris. The output is shown in Output 52.1.1.

```
options validvarname=any;
title 'World Weather Online Data for Paris';
LIBNAME myLib saserain "%sysget(RAIN_DATA)"
   OUTXML=gstart
   AUTOMAP=replace
   MAPREF=MyMap
   XMLMAP="%sysget(RAIN_DATA)gstart.map"
   APIKEY='XXXXXXXXXXXXXXXXXXX'
   QUERY='Paris,France'
   FORMAT=xml
   NUM_OF_DAYS=1;

data howCool;
   set myLib.gstart ;
run;

proc contents data=howCool; run;
proc print data=howCool(obs=6); run;
```

Output 52.1.1 World Weather Online Data for Paris

World Weather Online Data for Paris

Obs	date	AreaName	Country	Region	oc	latitude	longitude	maxtempC	maxtempF	mintempC
1	2016-10-18	Paris	France	Ile-de-France	1	48.8670	2.33300	16	60	7
2	2016-10-18	Paris	France	Ile-de-France	1	48.8670	2.33300	16	60	7
3	2016-10-18	Paris	France	Ile-de-France	1	48.8670	2.33300	16	60	7
4	2016-10-18	Paris	France	Ile-de-France	1	48.8670	2.33300	16	60	7

Obs	mintempF	uvIndex	time	tempC	tempF	windspeedMiles	windspeedKmph	winddirDegree	winddir16Point
1	45	2	0	10	50	6	10	238	WSW
2	45	2	600	9	48	7	11	243	WSW
3	45	2	1200	14	57	11	17	242	WSW
4	45	2	1800	14	56	13	20	267	W

Obs	weatherCode	weatherDesc	precipMM	humidity	visibility	pressure	cloudcover	HeatIndexC	HeatIndexF
1	116	Partly cloudy	0.00000	90	10	1024	15	10	49
2	113	Clear	0.00000	94	5	1024	17	8	46
3	116	Partly cloudy	0.00000	67	10	1024	46	15	60
4	176	Patchy rain possible	0.60000	79	9	1023	87	14	56

Obs	DewPointC	DewPointF	WindChillC	WindChillF	WindGustMiles	WindGustKmph	FeelsLikeC	FeelsLikeF
1	8	47	9	47	11	18	9	47
2	7	45	6	43	13	21	6	43
3	9	49	15	59	14	22	15	59
4	10	50	13	55	13	21	13	55

Obs	chanceofrain	chanceofremdry	chanceofwindy	chanceofovercast	chanceofsunshine	chanceoffrost
1	1	0	0	0	98	0
2	1	0	0	0	96	0
3	0	0	0	0	94	0
4	81	0	0	26	19	0

Obs	chanceofhightemp	chanceoffog	chanceofsnow	chanceofthunder
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

The SASERAIN interface engine supports the XML format. The XML data that the World Weather Online website returns are placed in a file named by the OUTXML= option (GSTART). The XML map that is automatically created is assigned the full pathname specified by the XMLMAP= option, and the fileref that is used for the map assignment is specified by the MAPREF= option. Because RAIN_DATA resolves to **U:\rain940\test**, the SASERAIN engine uses the MAPREF= and XMLMAP= options in the FILENAME statement to assign a filename:

```
FILENAME MyMap "U:\rain940\test\gstart.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 described in the section "SAS OUTXML File" on page 3658. The XML data file is placed

in the current working folder and the SAS data set that is created when the XML data are read into SAS is placed in the location specified by *physical-name,* which is described in the section "The LIBNAME *libref* SASERAIN Statement" on page 3651. You can refer to your data by using the myLib libref in your SASERAIN LIBNAME statement. The myLib libref is shown inside the DATA step in the SET statement. The SET statement reads observations from the input data set myLib.gstart and stores them in a SAS data set named HowCool, as shown in Figure 52.1.1. You can also use the SAS DATA step to perform further processing and to store the resulting time series in a SAS data set; this process is described in the section "SAS Output Data Set" on page 3658.

To specify the list of World Weather Online data sets that you want to retrieve, use the QUERY= option. This option accepts a string, enclosed in single quotation marks, that denotes a list of World Weather Online location codes that specify the places where you want the weather forecast data to be selected for the resulting SAS data set. The World Weather Online location codes are separated by semicolons, so valid World Weather Online codes cannot contain embedded semicolons or quotes. The HowCool data set contains the local weather forecast variables The observation range is controlled by the NUM_OF_DAYS= option, which is a required option. The HowCool data set contains observations that start today and end the same day, as specified by the NUM_OF_DAYS option. The frequency of the data is the six-hour default, because the FREQ= option is not specified.

NOTE: The "%20" is a special character for URL encoding of blanks. If the World Weather Online code that you name in the QUERY= option contains a blank, the SASERAIN engine uses "%20" wherever the blank appears in the World Weather Online code. If the World Weather Online code contains an underscore, then you must use an underscore in the QUERY= option. The underscore and the blank are not equivalent in World Weather Online databases.

Example 52.2: Retrieving the Two-Day Local Weather Forecast for One Location

The statements that follow enable you to access the weather for London for two days (NUM_OF_DAYS=2), which starts with today. The observations are given at a frequency of every 24 hours and are sorted in chronological order. The output is shown in Output 52.2.1.

```
options validvarname=any;
title 'Retrieve Two Day Weather Forecast for London';
libname _all_ clear;

libname mylib "U:\rain940\doc\";

libname rain saserain "%sysget(RAIN_DATA)"
   QUERY='London,United Kingdom'
   OUTXML=foggy
   XMLMAP="%sysget(RAIN_DATA)foggy.map"
   APIKEY='XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
   NUM_OF_DAYS=2
   TP=24
   FORMAT=xml;

data mylib.london_fog;
   set rain.foggy;
```

```
run;

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

Output 52.2.1 London Weather for Today and Tomorrow: London_fog

Retrieve Two Day Weather Forecast for London

Obs	date	AreaName	Country	Region	oc	latitude	longitude	maxtempC	maxtempF	mintempC
1	2016-10-18	London	United Kingdom	City of London, Greater London	1	51.5170	-0.106	14	57	6
2	2016-10-19	London	United Kingdom	City of London, Greater London	1	51.5170	-0.106	14	57	7

Obs	mintempF	uvIndex	time	tempC	tempF	windspeedMiles	windspeedKmph	winddirDegree	winddir16Point
1	43	2	24	14	57	14	23	284	WNW
2	44	2	24	14	57	11	18	311	NW

Obs	weatherCode	weatherDesc	precipMM	humidity	visibility	pressure	cloudcover	HeatIndexC	HeatIndexF
1	113	Sunny	1.30000	79	10	1020	5	13	55
2	119	Cloudy	0.00000	79	10	1022	67	14	57

Obs	DewPointC	DewPointF	WindChillC	WindChillF	WindGustMiles	WindGustKmph	FeelsLikeC	FeelsLikeF
1	5	40	11	51	18	30	11	51
2	8	46	12	53	16	26	12	53

Obs	chanceofrain	chanceofremdry	chanceofwindy	chanceofovercast	chanceofsunshine	chanceoffrost
1	95	0	0	2	94	0
2	2	0	0	3	0	0

Obs	chanceofhightemp	chanceoffog	chanceofsnow	chanceofthunder
1	0	0	0	0
2	0	0	0	0

The XML data that the World Weather Online website returns are placed in a file that is named by the OUTXML= option—in this case, *FOGGY1.xml*. **NOTE:** The SASERAIN engine appends a numeral to the XML filename, and the file extension (.xml) is excluded from the filename that appears in the OUTXML= option. The SAS data set created when the XML data file is read into SAS is placed in the location that is specified inside the string enclosed in double quotation marks in the SASERAIN LIBNAME statement. So, in the preceding example, if the RAIN_DATA environment variable is set to **U:\rain940\test**, then the downloaded XML file is located at **U:\rain940\test\FOGGY1.xml**.

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

```
libname rain saserain "U:\rain940\test\"
   OUTXML=foggy
   XMLMAP="U:\rain940\test\foggy.map"
   APIKEY='XXXXXXXXXXXXXXXXXXXX'
   QUERY='London,United Kingdom';
```

You could also use either a SAS macro variable or a system environment variable to store the value of your World Weather Online 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 SASERAIN engine appends a numeral to the XML filename to indicate the position of the World Weather Online location code in the QUERY= option.

The QUERY= option specifies the list of World Weather Online locations that you want to retrieve weather data for. This option accepts a string, enclosed in single quotation marks, that consists of one or more World Weather Online locations that you select (keep) in the resulting SAS data set. The result, FOGGY, is named in the DATA step and is shown in Figure 52.2.1. The preceding example uses only one World Weather Online code, which is in the first position of the QUERY= option, so the numeral 1 is appended to the name of the XML file, resulting in *FOGGY1.xml*.

It is more efficient to use the DATA step to store your World Weather Online 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 SASERAIN libref directly, as in the statement

```
proc print data=rain.foggy;
```

This statement uses the member name, FOGGY, in the PROC PRINT statement; this usage corresponds to specifying the OUTXML=FOGGY option. Although using this statement might seem easier, it is not as efficient, because every time you use the SASERAIN libref, the SASERAIN 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.

Example 52.3: Retrieving the Local Weather Forecast for One Location

This example shows how to use one World Weather Online location query to retrieve weather data for Dubai, starting today and ending tomorrow (num_of_days=2), with a 24-hour frequency. The output is shown in Output 52.3.1.

```
options validvarname=any;

title 'Retrieve Weather Data for Dubai';
libname _all_ clear;
libname mylib "U:\rain940\doc\";

libname myplace saserain "%sysget(RAIN_DATA)"
   apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
   query='Dubai,United Arab Emirates'
   format=XML
   outXml=dubhot
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(RAIN_DATA)dubhot.map"
   num_of_days=2
   tp=24
   ;

data mylib.hotdub;
   set myplace.dubhot;
run;

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

Output 52.3.1 Local Weather for Dubai

Retrieve Weather Data for Dubai

Obs	date	AreaName	Country	Region	oc	latitude	longitude	maxtempC	maxtempF	mintempC	mintempF
1	2016-10-18	Dubai	United Arab Emirates	Dubai	1	25.2520	55.2800	38	100	28	83
2	2016-10-19	Dubai	United Arab Emirates	Dubai	1	25.2520	55.2800	37	99	28	83

Obs	uvIndex	time	tempC	tempF	windspeedMiles	windspeedKmph	winddirDegree	winddir16Point
1	8	24	38	100	7	12	203	SSW
2	8	24	37	99	8	13	278	W

Obs	weatherCode	weatherDesc	precipMM	humidity	visibility	pressure	cloudcover	HeatIndexC	HeatIndexF
1	113	Sunny	0	51	10	1011	2	43	109
2	113	Sunny	0	50	10	1011	0	44	111

Obs	DewPointC	DewPointF	WindChillC	WindChillF	WindGustMiles	WindGustKmph	FeelsLikeC	FeelsLikeF
1	23	73	36	97	12	19	43	109
2	23	73	37	99	12	20	44	111

Obs	chanceofrain	chanceofremdry	chanceofwindy	chanceofovercast	chanceofsunshine	chanceoffrost
1	1	0	0	0	96	0
2	1	0	0	0	100	0

Obs	chanceofhightemp	chanceoffog	chanceofsnow	chanceofthunder
1	97	0	0	0
2	93	0	0	0

Example 52.4: Retrieving the Local Weather Forecast for Three Locations

This example shows how to retrieve World Weather Online data for three locations (London, Paris, and Dubai), starting today and ending today (num_of_days=1), with a 24-hour frequency. The output is shown in Output 52.4.1.

```
options validvarname=any;

title 'Retrieve Weather Data for Three Cities';
libname _all_ clear;
libname mylib "U:\rain940\doc\";

libname rain saserain "%sysget(RAIN_DATA)"
   apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
   query='London,United Kingdom;Paris,France;Dubai,United Arab Emirates'
   format=XML
   outXml=tricity
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(RAIN_DATA)tricity.map"
   num_of_days=1
   tp=24
   ;

data mylib.threecit;
   set rain.tricity;
run;

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

Output 52.4.1 Local Weather for London, Paris, and Dubai

Retrieve Weather Data for Three Cities

Obs	date	AreaName	Country	Region	oc	latitude	longitude	maxtempC	maxtempF	mintempC
1	2016-10-18	London	United Kingdom	City of London, Greater London	1	51.5170	-0.1060	14	57	6
2	2016-10-18	Paris	France	Ile-de-France	2	48.8670	2.3330	16	60	7
3	2016-10-18	Dubai	United Arab Emirates	Dubai	3	25.2520	55.2800	38	100	28

Obs	mintempF	uvIndex	time	tempC	tempF	windspeedMiles	windspeedKmph	winddirDegree	winddir16Point
1	43	2	24	14	57	14	23	284	WNW
2	45	2	24	16	60	9	15	244	WSW
3	83	8	24	38	100	7	12	203	SSW

Obs	weatherCode	weatherDesc	precipMM	humidity	visibility	pressure	cloudcover	HeatIndexC	HeatIndexF
1	113	Sunny	1.30000	79	10	1020	5	13	55
2	116	Partl	0.90000	83	10	1024	52	17	63
3	113	Sunny	0.00000	51	10	1011	2	43	109

Obs	DewPointC	DewPointF	WindChillC	WindChillF	WindGustMiles	WindGustKmph	FeelsLikeC	FeelsLikeF
1	5	40	11	51	18	30	11	51
2	9	49	17	63	13	21	17	63
3	23	73	36	97	12	19	43	109

Obs	chanceofrain	chanceofremdry	chanceofwindy	chanceofovercast	chanceofsunshine	chanceoffrost
1	95	0	0	2	94	0
2	92	0	0	0	94	0
3	1	0	0	0	96	0

Obs	chanceofhightemp	chanceoffog	chanceofsnow	chanceofthunder
1	0	0	0	0
2	0	0	0	0
3	97	0	0	0

Example 52.5: Retrieving Current Conditions for One Location

This example shows how to retrieve current conditions data for one location, Paris. Output 52.5.1 shows the current weather conditions data.

```
title 'Current Conditions for Paris';
libname _all_ clear;
options validvarname=any;
libname mylib "U:\rain940\doc\";

libname myRain saserain "%sysget(RAIN_DATA)"
   apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
   query='Paris,France'
   num_of_days=1
   conditions=onlycc
   outxml=parcon
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(RAIN_DATA)parcon.map"
   format=xml
   ;

data mylib.parcon;
   set myRain.parcon;
run;


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

Output 52.5.1 Local Current Weather Conditions for Paris

Current Conditions for Paris

Obs	AreaName	Country	Region	observation_time	oc	latitude	longitude	temp_C	temp_F
1	Paris	France	Ile-de-France	14:11:00	1	48.867	2.333	16	61

Obs	weatherCode	weatherDesc	windspeedMiles	windspeedKmph	winddirDegree	winddir16Point
1	116	Partly cloudy	16	26	250	WSW

Obs	precipMM	humidity	visibility	pressure	cloudcover	FeelsLikeC	FeelsLikeF
1	0	63	10	1023	0	16	61

Example 52.6: Retrieving Historical Weather Data for Two Cities for a Date Range

This example shows how to retrieve past weather data for two locations (London and Paris) by using a date range. The historical (past) weather API is invoked because the DATE= and ENDDATE= options are specified. The concept of current conditions does not have any meaning when you specify past dates, so the historical weather data are returned instead of the current conditions. The output is shown in Output 52.6.1. When you specify past dates, the same data are returned whether or not you specify the CC= option. The SAS log shows the following warning:

```
*****WARNING: Using historical (past) weather API, so current conditions are
not reported.
```

```
options validvarname=any;

title 'Historical Weather for Date Range SEPT 07, 2016 - SEPT 08, 2016 for
   London and Paris';
libname _all_ clear;
libname mylib "U:\rain940\doc\";

libname myRain saserain "%sysget(RAIN_DATA)"
   apikey='XXXXXXXXXXXXXXXXXXXX'
   query='London,United Kingdom;Paris,France'
   date='2016-09-07'
   enddate='2016-09-08'
   tp=24
   cc=onlycc
   format=XML
   outXml=rainex05
   automap=replace
   mapref=MyMap
   xmlmap="%sysget(RAIN_DATA)rainex05.map"
   ;

data mylib.cc3day;
   set myRain.rainex05;
run;

proc contents data=mylib.cc3day; run;
proc print data=mylib.cc3day; run;
```

Output 52.6.1 Historical Weather Data for Date Range for London and Paris

Historical Weather for Date Range SEPT 07, 2016 - SEPT 08, 2016 for London and Paris

Obs	date	AreaName	Country	Region	oc	latitude	longitude	maxtempC	maxtempF	mintempC
1	2016-09-07	London	United Kingdom	City of London, Greater London	1	51.5170	-0.10600	27	80	16
2	2016-09-07	Paris	France	Ile-de-France	2	48.8670	2.33300	27	81	17
3	2016-09-08	London	United Kingdom	City of London, Greater London	1	51.5170	-0.10600	24	74	15
4	2016-09-08	Paris	France	Ile-de-France	2	48.8670	2.33300	26	80	14

Obs	mintempF	uvIndex	time	tempC	tempF	windspeedMiles	windspeedKmph	winddirDegree	winddir16Point
1	61	0	24	27	80	11	18	157	SSE
2	63	0	24	27	81	9	14	93	E
3	59	0	24	24	74	17	27	266	W
4	56	0	24	26	80	14	22	290	WNW

Obs	weatherCode	weatherDesc	precipMM	humidity	visibility	pressure	cloudcover	HeatIndexC	HeatIndexF
1	119	Cloudy	0	66	10	1018	81	28	82
2	113	Sunny	0	43	10	1018	0	28	83
3	113	Sunny	0	56	10	1010	1	25	76
4	113	Sunny	0	61	10	1013	23	27	80

Obs	DewPointC	DewPointF	WindChillC	WindChillF	WindGustMiles	WindGustKmph	FeelsLikeC	FeelsLikeF
1	19	67	26	79	10	15	28	82
2	14	58	28	83	11	17	28	83
3	13	55	22	71	18	29	25	76
4	17	63	25	78	16	26	27	80

References

World Weather Online (2016). "World Weather Online API for Developers and Programmers, Local City and Town Weather API." Accessed August 4, 2016. http://www.worldweatheronline.com/api/docs/local-city-town-weather-api.aspx.

Subject Index

Syntax Index