Chapter 53
The SASERAIN Interface Engine

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 15 days of premium local weather forecast data. The SASERAIN engine supports only the premium local
weather API because World Weather Online has discontinued the nonpremium API. The default range for the SASERAIN engine is 2 days. You can use the premium local weather forecast API if you subscribe to the premium service and also specify your premium API key. The premium API key provides a maximum date range of 15 days.

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 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 premium API key (authentication token). To obtain your own unique API key, visit the World Weather Online website at the following URL:

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


The World Weather Online API key is a 31-character mixed-case alphanumeric string, such as “abCDefghi-jklmnopqrstuvwxyz123456789,” 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 3703 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.

In the following example, and “Examples: SASERAIN Interface Engine” on page 3703, use the SAS option SSLCALISTLOC=<specify the location of your CA certificates here>. The specification shown as SSLCALISTLOC= ／SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem” is for demonstration purposes only. Specify your own location of your trusted certificates inside the double quotes.

The statements that follow enable you to access 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 53.1.

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

title 'Retrieve Weather Data for London, Paris, and Dubai';
libname _all_ clear;
libname mylib "\sasusr\playpens\saskff\rain\doc\";
libname rain saserain "\sasusr\playpens\saskff\rain\test\"
  QUERY='London,United Kingdom;Paris,France;Dubai,United Arab Emirates'
  FX24=yes
  CONDITIONS=yes
  OUTXML=tricky
  AUTOMAP=replace
  MAPREF=MyMap
  XLMAP="\sasusr\playpens\saskff\rain\test\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 "\sasusr\playpens\saskff\rain\test\";
```
data mylib.mycca;
   set condo.cc_tricky;
run;

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

**Figure 53.1** Current Conditions for London, Paris, and Dubai

**Retrieve Weather Data for London, Paris, and Dubai**

<table>
<thead>
<tr>
<th>Obs</th>
<th>AreaName</th>
<th>Country</th>
<th>Region</th>
<th>weatherDesc</th>
<th>winddir16Point</th>
<th>observation_time</th>
<th>oc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>London</td>
<td>United Kingdom</td>
<td>City of London, Greater London</td>
<td>Overcast</td>
<td>NE</td>
<td>13:45:00</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>Sunny</td>
<td>ENE</td>
<td>13:45:00</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Dubai</td>
<td>United Arab Emirates</td>
<td>Dubai</td>
<td>Sunny</td>
<td>W</td>
<td>13:45:00</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>latitude</th>
<th>longitude</th>
<th>temp_C</th>
<th>temp_F</th>
<th>weatherCode</th>
<th>windspeedMiles</th>
<th>windspeedKmph</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51.517</td>
<td>-0.106</td>
<td>12</td>
<td>54</td>
<td>122</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>48.867</td>
<td>2.333</td>
<td>16</td>
<td>61</td>
<td>113</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>25.252</td>
<td>55.280</td>
<td>36</td>
<td>97</td>
<td>113</td>
<td>11</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>winddirDegree</th>
<th>precipMM</th>
<th>humidity</th>
<th>visibility</th>
<th>pressure</th>
<th>cloudcover</th>
<th>FeelsLikeC</th>
<th>FeelsLikeF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>0.1</td>
<td>54</td>
<td>10</td>
<td>1021</td>
<td>100</td>
<td>11</td>
<td>51</td>
</tr>
<tr>
<td>2</td>
<td>70</td>
<td>0.0</td>
<td>42</td>
<td>10</td>
<td>1017</td>
<td>0</td>
<td>16</td>
<td>61</td>
</tr>
<tr>
<td>3</td>
<td>270</td>
<td>0.0</td>
<td>21</td>
<td>10</td>
<td>1006</td>
<td>0</td>
<td>35</td>
<td>94</td>
</tr>
</tbody>
</table>

The XML data that the World Weather Online website returns are placed in a file that is named by the OUT-XML= 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.

You could 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 53.1. 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 53.1.

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

```sas
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 3703.

### 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 53.1 summarizes the options that the SASERAIN engine uses. In addition, there is one required option: APIKEY='rain_api_key'.

---

**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 53.1 summarizes the options that the SASERAIN engine uses. In addition, there is one required option: APIKEY='rain_api_key'.

---
**Table 53.1 Summary of LIBNAME libref SASERAIN Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIKEY=</td>
<td>Specifies the required World Weather Online access key that enables you to access the data that the World Weather Online website provides.</td>
</tr>
<tr>
<td>AUTOMAP=</td>
<td>Specifies whether or not to overwrite the existing XML map file.</td>
</tr>
<tr>
<td>CONDITIONS=</td>
<td>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.</td>
</tr>
<tr>
<td>CONNECT=</td>
<td>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.</td>
</tr>
<tr>
<td>DATE=</td>
<td>Specifies the beginning date for past weather data for the specified range: specify the start date in 'YYYY-MM-DD' format.</td>
</tr>
<tr>
<td>DAY=</td>
<td>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.</td>
</tr>
<tr>
<td>DEBUG=</td>
<td>Specifies whether or not to include diagnostic message logging in the SAS log window.</td>
</tr>
<tr>
<td>ENDDATE=</td>
<td>Specifies the end date for past weather for the specified range: specify the end date in 'YYYY-MM-DD' format. The end date must have the same month and year as the DATE= option.</td>
</tr>
<tr>
<td>FORECAST=</td>
<td>Specifies whether or not to return the weather forecast for a given postal code, zip code, and latitude/longitude values.</td>
</tr>
<tr>
<td>FORMAT=</td>
<td>Specifies a file extension that indicates the type of file to retrieve. Only XML is supported for the SASERAIN interface engine.</td>
</tr>
<tr>
<td>FREQ=</td>
<td>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.</td>
</tr>
<tr>
<td>FX24=</td>
<td>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).</td>
</tr>
<tr>
<td>NUM_OF_DAYS=</td>
<td>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 2 days of forecast data, and the maximum is 15 days (premium weather API).</td>
</tr>
</tbody>
</table>
### Table 53.1 continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OUTCC=</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>OUTXML=</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>PROXY=</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>QUERY=</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>TP=</strong></td>
<td>Specifies the time period (interval) of the selected weather forecast data in number of hours: 1, 3, 6 (default), 12, or 24 hours.</td>
</tr>
<tr>
<td><strong>XMLMAP=</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

---

**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 53.2.

**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 53.2 Current Conditions Forecast Variables

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

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 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 earliest start date (which you specify in the 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 2 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 53.3. 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>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Local forecast date in 'YYYY-MM-DD' format. For example: 2013-05-31.</td>
</tr>
<tr>
<td>maxtempC</td>
<td>Maximum temperature of the day in degrees Celsius</td>
</tr>
<tr>
<td>maxtempF</td>
<td>Maximum temperature of the day in degrees Fahrenheit</td>
</tr>
<tr>
<td>mintempC</td>
<td>Minimum temperature of the day in degrees Celsius</td>
</tr>
<tr>
<td>mintempF</td>
<td>Minimum temperature of the day in degrees Fahrenheit</td>
</tr>
<tr>
<td>uvIndex</td>
<td>Ultraviolet radiation index</td>
</tr>
<tr>
<td>time</td>
<td>Local time in 'hmm' format. For example: 100 or 1500.</td>
</tr>
<tr>
<td>tempC</td>
<td>Temperature in degrees Celsius</td>
</tr>
<tr>
<td>tempF</td>
<td>Temperature in degrees Fahrenheit</td>
</tr>
<tr>
<td>windspeedMiles</td>
<td>Wind speed in miles per hour</td>
</tr>
<tr>
<td>windspeedKmph</td>
<td>Wind speed in kilometers per hour</td>
</tr>
<tr>
<td>windspeedKnots</td>
<td>Wind speed in knots</td>
</tr>
<tr>
<td>windspeedMeterSec</td>
<td>Wind speed in meters per second</td>
</tr>
<tr>
<td>winddirDegree</td>
<td>Wind direction in degrees</td>
</tr>
<tr>
<td>winddir16Point</td>
<td>Wind direction on a 16-point compass</td>
</tr>
<tr>
<td>weatherCode</td>
<td>Weather condition code</td>
</tr>
<tr>
<td>weatherDesc</td>
<td>Weather condition description</td>
</tr>
<tr>
<td>weatherIconUrl</td>
<td>URL for weather icon</td>
</tr>
<tr>
<td>precipMM</td>
<td>Precipitation in millimeters</td>
</tr>
<tr>
<td>precipinches</td>
<td>Precipitation in inches</td>
</tr>
<tr>
<td>humidity</td>
<td>Humidity in percentage</td>
</tr>
<tr>
<td>visibility</td>
<td>Visibility in kilometers</td>
</tr>
<tr>
<td>visibilityMiles</td>
<td>Visibility in miles</td>
</tr>
<tr>
<td>pressure</td>
<td>Atmospheric pressure in millibars</td>
</tr>
<tr>
<td>pressureInches</td>
<td>Atmospheric pressure in inches</td>
</tr>
<tr>
<td>cloudcover</td>
<td>Cloud cover in percentage</td>
</tr>
<tr>
<td>chanceofrain</td>
<td>Chance of rain (precipitation) in percentage</td>
</tr>
<tr>
<td>chanceofwindy</td>
<td>Chance of being windy in percentage</td>
</tr>
<tr>
<td>chanceofovercast</td>
<td>Chance of being cloudy in percentage</td>
</tr>
<tr>
<td>chanceofsunny</td>
<td>Chance of being sunny in percentage</td>
</tr>
<tr>
<td>chanceoffrost</td>
<td>Chance of frost in percentage</td>
</tr>
<tr>
<td>chanceoffog</td>
<td>Chance of fog in percentage</td>
</tr>
<tr>
<td>chanceofsnow</td>
<td>Chance of snow in percentage</td>
</tr>
<tr>
<td>chanceofthunder</td>
<td>Chance of thunder in percentage</td>
</tr>
</tbody>
</table>
The LIBNAME libref SASERAIN Statement

**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 3703:

```
FILENAME MyMap "/sasusr/playpens/saskff/rain/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 3703, 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 15 days.

**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.

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.
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 you obtain by using the search API on the web page with the following URL:

https://api.worldweatheronline.com/premium/v1/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 three 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, ’XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX’, 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:

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 3695.

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.
Example 53.1: 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 53.1.1.

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

title 'World Weather Online Data for Paris';
LIBNAME myLib saserain '/sasusr/playpens/saskff/rain/test/
   OUTXML=gstart
   AUTOMAP=replace
   MAPREF=MyMap
   XMLMAP='/sasusr/playpens/saskff/rain/test/gstart.map'
   APIKEY='XXXXXXXXXXXXXXXXXXXXX'
   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;
```
Chapter 53: The SASERAIN Interface Engine

Output 53.1.1 World Weather Online Data for Paris

World Weather Online Data for Paris

<table>
<thead>
<tr>
<th>Obs</th>
<th>date</th>
<th>AreaName</th>
<th>Country</th>
<th>Region</th>
<th>oc</th>
<th>latitude</th>
<th>longitude</th>
<th>maxtempC</th>
<th>maxtempF</th>
<th>mintempC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2017-05-09</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>1</td>
<td>48.8670</td>
<td>2.33300</td>
<td>15</td>
<td>59</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2017-05-09</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>1</td>
<td>48.8670</td>
<td>2.33300</td>
<td>15</td>
<td>59</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>2017-05-09</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>1</td>
<td>48.8670</td>
<td>2.33300</td>
<td>15</td>
<td>59</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>2017-05-09</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>1</td>
<td>48.8670</td>
<td>2.33300</td>
<td>15</td>
<td>59</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>mintempF</th>
<th>totalSnow_cm</th>
<th>sunHour</th>
<th>uIndex</th>
<th>time</th>
<th>tempC</th>
<th>tempF</th>
<th>windspeedMiles</th>
<th>windspeedKmph</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>43</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>50</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>43</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>50</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>50</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>43</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>50</td>
<td>8</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>winddirDegree</th>
<th>winddir16Point</th>
<th>weatherCode</th>
<th>weatherDesc</th>
<th>precipMM</th>
<th>humidity</th>
<th>visibility</th>
<th>pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>NNE</td>
<td>176</td>
<td>Patchy rain possible</td>
<td>0</td>
<td>76</td>
<td>19</td>
<td>1023</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>NNE</td>
<td>113 Clear</td>
<td></td>
<td>0</td>
<td>83</td>
<td>19</td>
<td>1022</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>NE</td>
<td>113 Sunny</td>
<td></td>
<td>0</td>
<td>61</td>
<td>19</td>
<td>1023</td>
</tr>
<tr>
<td>4</td>
<td>52</td>
<td>NE</td>
<td>113 Sunny</td>
<td></td>
<td>0</td>
<td>43</td>
<td>20</td>
<td>1018</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>cloudcover</th>
<th>HeatIndexC</th>
<th>HeatIndexF</th>
<th>DewPointC</th>
<th>DewPointF</th>
<th>WindChillC</th>
<th>WindChillF</th>
<th>WindGustMiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>93</td>
<td>11</td>
<td>51</td>
<td>7</td>
<td>45</td>
<td>11</td>
<td>51</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>8</td>
<td>46</td>
<td>5</td>
<td>42</td>
<td>6</td>
<td>43</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>13</td>
<td>56</td>
<td>6</td>
<td>43</td>
<td>13</td>
<td>56</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>16</td>
<td>61</td>
<td>3</td>
<td>38</td>
<td>16</td>
<td>61</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>WindGustKmph</th>
<th>FeelsLikeC</th>
<th>FeelsLikeF</th>
<th>chanceofrain</th>
<th>chanceofremdry</th>
<th>chanceofwindy</th>
<th>chanceofovercast</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28</td>
<td>11</td>
<td>51</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>6</td>
<td>43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>13</td>
<td>56</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>16</td>
<td>61</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>chanceofsunshine</th>
<th>chanceoffrost</th>
<th>chanceofhightemp</th>
<th>chanceoffog</th>
<th>chanceofsnow</th>
<th>chanceofthunder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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 the XMLMAP= option is specified as /sasusr/playpens/saskff/rain/test/, the SASERAIN engine uses the MAPREF= and XMLMAP= options in the FILENAME statement to assign a filename:

FILENAME MyMap "/sasusr/playpens/saskff/rain/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 3702. 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 3695. 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 53.1.1. 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 3702.

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 53.2: 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 53.2.1**.

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

title 'Retrieve Two Day Weather Forecast for London';
libname _all_ clear;

libname mylib "/sasusr/playpens/saskff/rain/doc/"
libname rain saserain "/sasusr/playpens/saskff/rain/test/"
   QUERY='London,United Kingdom'
   OUTXML=foggy
   XMLElement="/sasusr/playpens/saskff/rain/test/foggy.map"
   APIKEY='XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
   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 53.2.1  London Weather for Today and Tomorrow: London_fog

## Retrieve Two Day Weather Forecast for London

<table>
<thead>
<tr>
<th>Obs</th>
<th>date</th>
<th>AreaName</th>
<th>Country</th>
<th>Region</th>
<th>oc</th>
<th>latitude</th>
<th>longitude</th>
<th>maxtempC</th>
<th>maxtempF</th>
<th>mintempC</th>
<th>mintempF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2017-05-09</td>
<td>London</td>
<td>United Kingdom</td>
<td>City of London, Greater London</td>
<td>1</td>
<td>51.517</td>
<td>-0.106</td>
<td>14</td>
<td>58</td>
<td>6</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>2017-05-10</td>
<td>London</td>
<td>United Kingdom</td>
<td>City of London, Greater London</td>
<td>1</td>
<td>51.517</td>
<td>-0.106</td>
<td>15</td>
<td>60</td>
<td>7</td>
<td>44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>totalSnow_cm</th>
<th>sunHour</th>
<th>uvIndex</th>
<th>timeC</th>
<th>tempC</th>
<th>tempF</th>
<th>windspeedMiles</th>
<th>windspeedKmph</th>
<th>winddirDegree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>24</td>
<td>14</td>
<td>58</td>
<td>8</td>
<td>12</td>
<td>58</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>24</td>
<td>15</td>
<td>60</td>
<td>8</td>
<td>13</td>
<td>92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>winddir16Point</th>
<th>weatherCode</th>
<th>weatherDesc</th>
<th>precipMM</th>
<th>humidity</th>
<th>visibility</th>
<th>pressure</th>
<th>cloudcover</th>
<th>HeatIndexC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENE</td>
<td>113</td>
<td>Sunny</td>
<td>0</td>
<td>78</td>
<td>16</td>
<td>1024</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>E</td>
<td>116</td>
<td>Partly cloudy</td>
<td>0</td>
<td>79</td>
<td>20</td>
<td>1010</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>HeatIndexF</th>
<th>DewPointC</th>
<th>DewPointF</th>
<th>WindChilC</th>
<th>WindChillF</th>
<th>WindGustMiles</th>
<th>WindGustKmph</th>
<th>FeelsLikeC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49</td>
<td>6</td>
<td>42</td>
<td>8</td>
<td>46</td>
<td>11</td>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>51</td>
<td>7</td>
<td>44</td>
<td>9</td>
<td>48</td>
<td>11</td>
<td>18</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>FeelsLikeF</th>
<th>chanceofrain</th>
<th>chanceofrxdry</th>
<th>chanceofwindy</th>
<th>chanceofovercast</th>
<th>chanceofsunshine</th>
<th>chanceoffrost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>46</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>67</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>95</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>chanceofnightemp</th>
<th>chanceoffog</th>
<th>chanceofsnow</th>
<th>chanceofthunder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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.

You could 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 53.2.1. 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

```sas
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 53.3: 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 53.3.1.

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

title 'Retrieve Weather Data for Dubai';
libname _all_ clear;
libname mylib "'/sasusr/playpens/saskff/rain/doc/";

libname myplace saserain "'/sasusr/playpens/saskff/rain/test/"
   apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
   query='Dubai,United Arab Emirates'
   format=XML
   outXml=dubhot
   automap=replace
   mapref=MyMap
   xmlmap="/sasusr/playpens/saskff/rain/test/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 53.3.1 Local Weather for Dubai

#### Retrieve Weather Data for Dubai

<table>
<thead>
<tr>
<th>Obs</th>
<th>date</th>
<th>AreaName</th>
<th>Country</th>
<th>Region</th>
<th>oc</th>
<th>latitude</th>
<th>longitude</th>
<th>maxtempC</th>
<th>maxtempF</th>
<th>mintempC</th>
<th>mintempF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2017-05-09</td>
<td>Dubai</td>
<td>United Arab Emirates</td>
<td>Dubai</td>
<td>1</td>
<td>25.2520</td>
<td>55.2800</td>
<td>38</td>
<td>101</td>
<td>30</td>
<td>86</td>
</tr>
<tr>
<td>2</td>
<td>2017-05-10</td>
<td>Dubai</td>
<td>United Arab Emirates</td>
<td>Dubai</td>
<td>1</td>
<td>25.2520</td>
<td>55.2800</td>
<td>37</td>
<td>99</td>
<td>28</td>
<td>83</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>totalSnow_cm</th>
<th>sunHour</th>
<th>uvIndex</th>
<th>time</th>
<th>tempC</th>
<th>tempF</th>
<th>windspeedMiles</th>
<th>windspeedKmph</th>
<th>winddirDegree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>24</td>
<td>38</td>
<td>101</td>
<td>8</td>
<td>13</td>
<td>251</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>24</td>
<td>37</td>
<td>99</td>
<td>10</td>
<td>16</td>
<td>248</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>winddir16Point</th>
<th>weatherCode</th>
<th>weatherDesc</th>
<th>precipMM</th>
<th>humidity</th>
<th>visibility</th>
<th>pressure</th>
<th>cloudcover</th>
<th>HeatIndexC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WSW</td>
<td>113</td>
<td>Sunny</td>
<td>0</td>
<td>43</td>
<td>18</td>
<td>1007</td>
<td>20</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>WSW</td>
<td>113</td>
<td>Sunny</td>
<td>0</td>
<td>56</td>
<td>20</td>
<td>1005</td>
<td>0</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>HeatIndexF</th>
<th>DewPointC</th>
<th>DewPointF</th>
<th>WindChillC</th>
<th>WindChillF</th>
<th>WindGustMiles</th>
<th>WindGustKmph</th>
<th>FeelsLikeC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>20</td>
<td>68</td>
<td>35</td>
<td>94</td>
<td>11</td>
<td>18</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>23</td>
<td>73</td>
<td>33</td>
<td>91</td>
<td>14</td>
<td>22</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>FeelsLikeF</th>
<th>chanceofrain</th>
<th>chanceofretdry</th>
<th>chanceofwindy</th>
<th>chanceofovercast</th>
<th>chanceofsunshine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>chanceoffrost</th>
<th>chanceofhightemp</th>
<th>chanceoffog</th>
<th>chanceofsnow</th>
<th>chanceofthunder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Example 53.4: 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 53.4.1.

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

title 'Retrieve Weather Data for Three Cities';
libname _all_ clear;
libname mylib "/_sasusr/playpens/saskff/rain/doc/";

libname rain saserain "/_sasusr/playpens/saskff/rain/test/"
    apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
    query='London,United Kingdom;Paris,France;Dubai,United Arab Emirates'
    format=XML
    outXml=tricity
    automap=replace
    mapref=MyMap
    xmlmap="/sasusr/playpens/saskff/rain/test/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;
```
### Example 53.4: Retrieving the Local Weather Forecast for Three Locations

#### Output 53.4.1 Local Weather for London, Paris, and Dubai

#### Retrieve Weather Data for Three Cities

| Obs | date     | AreaName | Country       | Region                      | oc | latitude | longitude | maxtempC | maxtempF | mintempC | mintempF | totalSnow_cm | sunHour | uvIndex | timeC | tempC | tempF | windspeedMiles | windspeedKmph | winddirDegree | winddir16Point | weatherCode | weatherDesc | precipMM | humidity | visibility | pressure | cloudcover | HeatIndexC | HeatIndexF | DewPointC | DewPointF | WindChillC | WindChillF | WindGustMiles | WindGustKmph | FeelsLikeC | FeelsLikeF | chanceofrain | chanceoffog | chanceofovercast | chanceofsunshine | chanceoffrost | chanceofhighfog | chanceofcloud | chanceofhail | chanceofthunder |
|-----|----------|----------|---------------|----------------------------|----|----------|-----------|----------|----------|----------|---------|----------|---------|---------------|----------|---------|-------|-------|-------|----------------|-------------|--------------|---------------|------------|-------------|---------|----------|-----------|-----------|------------|-----------|-----------|----------|----------|----------------|----------------|--------------|--------------|-----------|-----------|-----------|
| 1   | 2017-05-09 | London  | United Kingdom | City of London, Greater London | 1  | 51.5170  | -0.1060   | 14       | 58       | 6        |         |         |         |               |          |         |       |       |       |                |             |              |              |            |             |         |          |           |           |            |          |          |         |         |                |                |              |              |          |           |           |          |
| 2   | 2017-05-09 | Paris   | France        | Île-de-France              | 2  | 48.8670  | 2.3330    | 15       | 59       | 6        |         |         |         |               |          |         |       |       |       |                |             |              |              |            |             |         |          |           |           |            |          |          |         |         |                |                |              |              |          |           |           |          |
| 3   | 2017-05-09 | Dubai   | United Arab Emirates | Dubai                 | 3  | 25.2520  | 55.2800   | 38       | 101      | 30       |         |         |         |               |          |         |       |       |       |                |             |              |              |            |             |         |          |           |           |            |          |          |         |         |                |                |              |              |          |           |           |          |
Example 53.5: Retrieving Current Conditions for One Location

This example shows how to retrieve current conditions data for one location, Paris. Output 53.5.1 shows the current weather conditions data.

```sas
%let apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXX';%libname myRain saserain '/sasusr/playpens/saskff/rain/test/'
%let query='Paris,France'
%let num_of_days=1
%let conditions=onlycc
%let outxml=parcon
%let automap=replace
%let mapref=MyMap
%let xmlmap="/sasusr/playpens/saskff/rain/test/parcon.map"
libname _all_ clear;
options validvarname=any
    sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";
libname mylib "'/sasusr/playpens/saskff/rain/doc/"; %libname myRain saserain "'/sasusr/playpens/saskff/rain/test/"
library
    %let apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
    %libname myRain saserain "'/sasusr/playpens/saskff/rain/test/"
    %let query='Paris,France'
    %let num_of_days=1
    %let conditions=onlycc
    %let outxml=parcon
    %let automap=replace
    %let mapref=MyMap
    %let xmlmap="/sasusr/playpens/saskff/rain/test/parcon.map"
    %format=xml
;
libname _all_ clear;
options validvarname=any
    sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";
libname mylib "'/sasusr/playpens/saskff/rain/doc/";
libname myRain saserain "'/sasusr/playpens/saskff/rain/test/"
library
    %let apikey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
    %libname myRain saserain "'/sasusr/playpens/saskff/rain/test/"
    %let query='Paris,France'
    %let num_of_days=1
    %let conditions=onlycc
    %let outxml=parcon
    %let automap=replace
    %let mapref=MyMap
    %let xmlmap="/sasusr/playpens/saskff/rain/test/parcon.map"
    %format=xml
;
data mylib.parcon;
    set myRain.parcon;
run;

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

---

**Output 53.5.1** Local Current Weather Conditions for Paris

**Current Conditions** for Paris

<table>
<thead>
<tr>
<th>Obs</th>
<th>AreaName</th>
<th>Country</th>
<th>Region</th>
<th>observation_time</th>
<th>oc</th>
<th>latitude</th>
<th>longitude</th>
<th>temp_C</th>
<th>temp_F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>13:45:00</td>
<td>1</td>
<td>48.867</td>
<td>2.333</td>
<td>16</td>
<td>61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>weatherCode</th>
<th>weatherDesc</th>
<th>windspeedMiles</th>
<th>windspeedKmph</th>
<th>winddirDegree</th>
<th>winddir16Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>113</td>
<td>Sunny</td>
<td>13</td>
<td>20</td>
<td>70</td>
<td>ENE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>precipMM</th>
<th>humidity</th>
<th>visibility</th>
<th>pressure</th>
<th>cloudcover</th>
<th>FeelsLikeC</th>
<th>FeelsLikeF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>42</td>
<td>10</td>
<td>1017</td>
<td>0</td>
<td>16</td>
<td>61</td>
</tr>
</tbody>
</table>
Example 53.6: 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 53.6.1. 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
         sslcalistloc="/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem";

title 'Historical Weather for Date Range MAY 01, 2017 - MAY 02, 2017 for London and Paris';
libname _all_ clear;
libname mylib "'/sasusr/playpens/saskff/rain/doc/";

libname myRain saserain "'/sasusr/playpens/saskff/rain/test/"
       apikey='XXXXXXXXXXXXXXXXXXXXXXXXX'
       query='London,United Kingdom;Paris,France'
       date='2017-05-01'
       enddate='2017-05-02'
       tp=24
       cc=onlycc
       format=XML
       outXml=rainex05
       automap=replace
       mapref=MyMap
       xmlmap="/sasusr/playpens/saskff/rain/test/rainex05.map"
;

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

proc contents data=mylib.cc3day; run;
proc print data=mylib.cc3day; run;
### Output 53.6.1 Historical Weather Data for Date Range for London and Paris

#### Historical Weather for Date Range MAY 01, 2017 - MAY 02, 2017 for London and Paris

<table>
<thead>
<tr>
<th>Obs</th>
<th>date</th>
<th>AreaName</th>
<th>Country</th>
<th>Region</th>
<th>date</th>
<th>latitude</th>
<th>longitude</th>
<th>maxtempC</th>
<th>maxtempF</th>
<th>mintempC</th>
<th>mintempF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2017-05-01</td>
<td>London</td>
<td>United Kingdom</td>
<td>City of London, Greater London</td>
<td>51.5170</td>
<td>-0.10600</td>
<td>15</td>
<td>59</td>
<td>8</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2017-05-01</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>48.8670</td>
<td>2.33300</td>
<td>15</td>
<td>59</td>
<td>8</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2017-05-02</td>
<td>London</td>
<td>United Kingdom</td>
<td>City of London, Greater London</td>
<td>51.5170</td>
<td>-0.10600</td>
<td>17</td>
<td>62</td>
<td>8</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2017-05-02</td>
<td>Paris</td>
<td>France</td>
<td>Ile-de-France</td>
<td>48.8670</td>
<td>2.33300</td>
<td>14</td>
<td>58</td>
<td>5</td>
<td>41</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>totalSnow_cm</th>
<th>sunHour</th>
<th>uvIndex</th>
<th>time</th>
<th>tempC</th>
<th>tempF</th>
<th>windspeedMiles</th>
<th>windspeedKmph</th>
<th>winddirDegree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>15</td>
<td>59</td>
<td>10</td>
<td>17</td>
<td>123</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>15</td>
<td>59</td>
<td>10</td>
<td>15</td>
<td>231</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>17</td>
<td>62</td>
<td>9</td>
<td>14</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>14</td>
<td>58</td>
<td>6</td>
<td>10</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>winddir16Point</th>
<th>weatherCode</th>
<th>weatherDesc</th>
<th>precipMM</th>
<th>humidity</th>
<th>visibility</th>
<th>pressure</th>
<th>cloudcover</th>
<th>HeatIndexC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ESE</td>
<td>353</td>
<td>Light rain shower</td>
<td>2.80000</td>
<td>85</td>
<td>9</td>
<td>1003</td>
<td>71</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>SW</td>
<td>176</td>
<td>Patchy rain possi</td>
<td>1.70000</td>
<td>67</td>
<td>10</td>
<td>1010</td>
<td>67</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>NE</td>
<td>353</td>
<td>Light rain shower</td>
<td>2.70000</td>
<td>82</td>
<td>10</td>
<td>1018</td>
<td>40</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>NE</td>
<td>176</td>
<td>Patchy rain possi</td>
<td>1.40000</td>
<td>66</td>
<td>9</td>
<td>1016</td>
<td>43</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>HeatIndexF</th>
<th>DewPointC</th>
<th>DewPointF</th>
<th>WindChillC</th>
<th>WindChillF</th>
<th>WindGustMiles</th>
<th>WindGustKmph</th>
<th>FeelsLikeC</th>
<th>FeelsLikeF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51</td>
<td>8</td>
<td>47</td>
<td>9</td>
<td>48</td>
<td>15</td>
<td>24</td>
<td>9</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>53</td>
<td>5</td>
<td>41</td>
<td>11</td>
<td>52</td>
<td>20</td>
<td>32</td>
<td>11</td>
<td>52</td>
</tr>
<tr>
<td>3</td>
<td>53</td>
<td>8</td>
<td>47</td>
<td>10</td>
<td>50</td>
<td>13</td>
<td>21</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>51</td>
<td>5</td>
<td>41</td>
<td>10</td>
<td>49</td>
<td>9</td>
<td>15</td>
<td>10</td>
<td>49</td>
</tr>
</tbody>
</table>

### References

Subject Index

APIKEY= option
   SASERAIN engine, 3696
AUTOMAP= option
   SASERAIN engine, 3696

CONDITIONS= option
   SASERAIN engine, 3696
CONNECT= option
   SASERAIN engine, 3697
creating a World Weather Online view, see SASERAIN engine

DATE= option
   SASERAIN engine, 3697
DAY= option
   SASERAIN engine, 3697
DEBUG= option
   SASERAIN engine, 3697
ENDDATE= option
   SASERAIN engine, 3697

FORECAST= option
   SASERAIN engine, 3698
FORMAT= option
   SASERAIN engine, 3699
FREQ= option
   SASERAIN engine, 3699
FX24= option
   SASERAIN engine, 3699

LIBNAME interface engine for World Weather Online data sets, see SASERAIN engine
LIBNAME libref SASERAIN statement
   SASERAIN engine, 3695
LIBNAME statement
   SASERAIN engine, 3689

MAPREF= option
   SASERAIN engine, 3699
MAPREF= option, SAS XML map
   SASERAIN engine, 3704

NUM_OF_DAYS= option
   SASERAIN engine, 3699

OUTCC= option
   SASERAIN engine, 3699
OUTXML= option
   OUTXML= option, SAS XML data
      SASERAIN engine, 3700
   SASERAIN engine, 3704
PROXY= option
   SASERAIN engine, 3700
QUERY= option
   SASERAIN engine, 3700

SAS output data set
   SASERAIN engine, 3702
SAS OUTXML file, XML file
   SASERAIN engine, 3702
SAS XML data, OUTXML= option
   SASERAIN engine, 3704
SAS XML format
   SASERAIN engine, 3704
SAS XML map file
   SASERAIN engine, 3702
SAS XML map, XMLMAP= option
   SASERAIN engine, 3704

SASERAIN engine
   APIKEY= option, 3696
   AUTOMAP= option, 3696
   CONDITIONS= option, 3696
   CONNECT= option, 3697
creating a World Weather Online view, 3689
   DATE= option, 3697
   DAY= option, 3697
   DEBUG= option, 3697
   ENDDATE= option, 3697
   FORECAST= option, 3698
   FORMAT= option, 3699
   FREQ= option, 3699
   FX24= option, 3699
   LIBNAME interface engine for World Weather Online data sets, 3689
   LIBNAME libref SASERAIN statement, 3695
   LIBNAME statement, 3689
   MAPREF= option, 3699
   MAPREF= option, SAS XML map, 3704
   NUM_OF_DAYS= option, 3699
   OUTCC= option, 3699
   OUTXML= option, 3700
   OUTXML= option, SAS XML data, 3704
   PROXY= option, 3700
   QUERY= option, 3700
   SAS output data set, 3702
   SAS OUTXML file, XML file, 3702
SAS XML data, OUTXML= option, 3704
SAS XML format, 3704
SAS XML map file, 3702
SAS XML map, XMLMAP= option, 3704
TP= option, 3701
viewing a World Weather Online data set, 3689
World Weather Online data files, 3689
World Weather Online Forecast data, 3689
XML format, 3704
XML map, MAPREF= option, 3704
XMLMAP= option, 3701
XMLMAP= option, SAS XML map, 3704

TP= option
SASERAIN engine, 3701
viewing a World Weather Online data set, see
SASERAIN engine
World Weather Online data files, see SASERAIN
engine
World Weather Online Forecast data, see SASERAIN
engine
XML format
SASERAIN engine, 3704
XML map, MAPREF= option
SASERAIN engine, 3704
XMLMAP= option
SASERAIN engine, 3701
XMLMAP= option, SAS XML map
SASERAIN engine, 3704
Syntax Index

APIKEY= option
  LIBNAME statement (SASERAIN), 3696
AUTOMAP= option
  LIBNAME statement (SASERAIN), 3696

CC= option
  LIBNAME statement (SASERAIN), 3696
CONNECT= option
  LIBNAME statement (SASERAIN), 3697

DAY= option
  LIBNAME statement (SASERAIN), 3697
DEBUG= option
  LIBNAME statement (SASERAIN), 3697

ENDDATE= option
  LIBNAME statement (SASERAIN), 3697

FORMAT= option
  LIBNAME statement (SASERAIN), 3699
FREQ= option
  LIBNAME statement (SASERAIN), 3699
FX24= option
  LIBNAME statement (SASERAIN), 3699
FX= option
  LIBNAME statement (SASERAIN), 3698

LIBNAME libref SASERAIN statement, 3695

MAPREF= option
  LIBNAME statement (SASERAIN), 3699

OUTCC= option
  LIBNAME statement (SASERAIN), 3699
OUTXML= option
  LIBNAME statement (SASERAIN), 3700

PROXY= option
  LIBNAME statement (SASERAIN), 3700
QUERY= option
  LIBNAME statement (SASERAIN), 3700

STDATE= option
  LIBNAME statement (SASERAIN), 3697

TP= option
  LIBNAME statement (SASERAIN), 3701
XMLMAP= option
  LIBNAME statement (SASERAIN), 3701