SAS/ETS® 14.2 User's Guide

The SASENOAA 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 50
The SASENOAA Interface Engine

Contents

Overview: SASENOAA Interface Engine

The SASENOAA interface engine enables SAS programmers to retrieve severe weather data from the National Oceanic and Atmospheric Administration (NOAA) Severe Weather Data Inventory (SWDI) web service, which is hosted jointly by the NOAA's National Environmental Satellite Data and Information Service (NESDIS), the US Department of Commerce National Climatic Data Center (NCDC), the University of North Carolina at Asheville's National Environmental Modeling and Analysis Center (NEMAC), and the Renaissance Computing Institute (RENCI) at UNC Asheville.

The SWDI web service offers access to severe weather data such as tornado vortex signatures; mesocyclone signatures; the digital mesocyclone detection algorithm; hail data; storm cell structure; preliminary local storm reports; and severe thunderstorm, tornado, flash flood, and special marine warnings. The SWDI lightning data are not accessible to the public, so they are not supported by the SASENOAA interface engine.

It is important to note that the absence of SWDI weather data for a geographic region or time period does not necessarily indicate that severe weather did not occur at that place or time; instead, the interpretation should be that severe weather was not detected or reported by NOAA's SWDI data sources. In addition, because much of the SWDI's information is derived from radar data, its usefulness is primarily that it provides data that indicates probable conditions for an event rather than confirming the actual occurrence of an event.

The SASENOAA interface engine uses the LIBNAME statement to enable you to specify how to retrieve your NOAA Severe Weather data and which weather data time series or storm events you want to retrieve based on date range and weather station 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 or in map files (such as Google Earth's KMZ files or Esri shapefiles). You can perform more analysis (if desired) either in the same SAS session or in a later session. You can map your results in Google Maps by importing the resulting KML file, or you can map your results in SAS by using PROC MAPIMPORT and PROC GMAP to create a map from the resulting Esri shapefiles (which have the filename extension .shp).

The SASENOAA interface engine is supported on SAS running on Linux X64 (LAX) and Windows. Although the SASENOAA engine uses the NOAA SWDI API, it is not endorsed or certified by either NOAA or the National Weather Service. By using the SASENOAA interface engine, you are agreeing to comply with the NOAA SWDI terms of use, which are described on the web page at the following URL:

http://www.weather.gov/disclaimer/

Getting Started: SASENOAA Interface Engine

You can query the SWDI data sets to retrieve the observations or data values for a list of time series or events by specifying the data format (FORMAT= option), the data set to access (NOAASET= option), and the date range (RANGE= option).

The SASENOAA engine's FORMAT= option supports three formats: XML (default), SHP, and KMZ. The XML format stores the weather data results in a SAS data set (.sas7bdat) named in the OUTXML= option, and when applicable, in two additional data sets: one for message text output (_M.sas7bdat), and another for statistics output (_S.sas7bdat).

The SHP format produces a ZIP file that contains four Esri shapefiles (with the extensions .shp, .shx, .dbf, and .prj). The SASENOAA interface unzips the SHP ZIP file to surface the four Esri files. The KMZ format produces a ZIP file (with the extension .kmz) that can be opened in virtual globe software such as Google Earth. The SASENOAA engine unzips the KMZ file to produce the resulting KML file, which can then be imported into Google Maps to create a detailed map of the SWDI time series data.

The NOAASET= option is required. You can specify one of the following Next-Generation Radar (NEXRAD) Level III types: nx3tvs (tornado vortex signatures), nx3meso (mesocyclone signatures), nx3mda (digital mesocyclone detection algorithm), nx3hail (hail signatures), or nx3structure (storm cell structure information). You can also specify two other types: plsr (preliminary local storm reports) and warn (severe thunderstorm, tornado, flash flood, and special marine warnings).

After you specify both the NOAASET= option and the format, you must also use the RANGE= option to specify the date range of the data that you are selecting for output, as shown in the following example.

The statements that follow enable you to access the severe weather tornado vortex signature (TVS) events that are recorded in the nx3tvs database for the date range beginning May 5, 2006, and ending May 6, 2006. The observations are sorted in chronological order (the datetime variable is ztime). The output is shown in Figure 50.1.

```
options validvarname=any;
title 'Retrieve Tornado Vortex Signature Data for the Range 20060505:20060506';
libname _all_ clear;

libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "%sysget(NOAA_DATA)"
   NOAASET=nx3tvs
   RANGE='20060505:20060506'
   OUTXML=cinco
   AUTOMAP=replace
   MAPREF=MyMap
   XMLMAP="U:\noaa940\test\cinco.map"
   FORMAT=xml;

data mylib.mycinco;
   set noaa.cinco;
run;


proc contents data=mylib.mycinco; run;
proc print data=mylib.mycinco(obs=10); run;
```

Figure 50.1 NX3TVS Data for May 5 to May 6, 2006

Retrieve Tornado Vortex Signature Data for the Range 20060505:20060506

Obs	ztime	wsr_id	cell_id	cell_type	range	azimuth	max_shear	mxdv	shape
1	2006-05-05T00:05:50	KBMX	Q0	TVS	7	217	403	116	POINT (-86.8535716274277 33.0786326913943)
2	2006-05-05T00:10:02	KBMX	Q0	TVS	5	208	421	120	POINT (-86.8165772540846 33.0982820681588)
3	2006-05-05T00:12:34	KSJT	P2	TVS	49	106	17	52	POINT (-99.5771091971025 31.1421609654838)
4	2006-05-05T00:17:31	KSJT	B4	TVS	40	297	25	62	POINT (-101.188161700093 31.672392833416)
5	2006-05-05T00:29:13	KMAF	H4	TVS	53	333	34	111	POINT (-102.664426480293 32.7306917937698)
6	2006-05-05T00:31:25	KLBB	N0	TVS	51	241	24	78	POINT (-102.70047613441 33.2380072329615)
7	2006-05-05T00:33:25	KMAF	H4	TVS	52	334	46	145	POINT (-102.6393683028 32.7226656893341)
8	2006-05-05T00:37:37	KMAF	H4	TVS	50	334	34	107	POINT (-102.621904684258 32.6927081076156)
9	2006-05-05T00:41:51	KMAF	H4	TVS	51	335	29	91	POINT (-102.614794815627 32.714139844846)
10	2006-05-05T00:44:33	KLBB	N0	TVS	46	245	35	100	POINT (-102.643380529494 33.3266446067682)

The XML data that the NOAA SWDI web service returns are placed in a file that is named by the OUTXML= option—in this case, *CINCO1.xml*. Note that the SASENOAA engine appends a numeral to the XML filename, and the file extension (.xml) is excluded from the filename that appears in the OUTXML= option. This XML data file resides in the location that is given inside the string enclosed in double quotation marks in the SASENOAA LIBNAME statement. So, in the preceding example, if the NOAA_DATA environment variable is set to **`U:\noaa940\test\`**, then the downloaded XML file is located at **`U:\noaa940\test\CINCO1.xml`**. An equivalent LIBNAME statement that does not use any environment variables could be as follows:

```
libname noaa sasenoaa "U:\noaa940\test\"
   NOAASET=nx3tvs
   RANGE='20060505:20060506'
   OUTXML=cinco
   XMLMAP="U:\noaa940\test\cinco.map"
   AUTOMAP=replace
   MAPREF=MyMap
   FORMAT=xml;
```

The XML map that is created is assigned the full pathname that the XMLMAP= option specifies. The SASENOAA engine appends a numeral to the XML filename to prevent filenames from being overwritten during multiple read requests.

The RANGE= option specifies the start date and end date for the range of days for which you want to retrieve data. This option accepts a string, enclosed in single quotation marks, that gives start and end dates (in `'YYYYMMDD'` format) so that only the recorded severe weather events from the selected dates are included. The result, MYCINCO, is named in the DATA step and is shown in Figure 50.1.

It is more efficient to use the DATA step to store your NOAA SWDI data in a SAS data set and then refer to the SAS data set directly in your PROC statements. You can also refer to the SASENOAA libref directly, as in the statement

```
proc print data=noaa.cinco(obs=10);
```

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

The SASENOAA interface engine supports the XML format by placing the XML data that the NOAA SWDI web service returns in a file named by the OUTXML= option. 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. In the preceding sample code, the SASENOAA engine uses the MAPREF= and XMLMAP= options in the FILENAME statement to assign a filename:

```
FILENAME MyMap "U:\noaa940\test\cinco.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 3599. The XML data file is placed in the folder that is designated by *physical-name,* which is described in the section "The LIBNAME *libref* SASENOAA Statement" on page 3587. You can refer to your data by using the NOAA libref defined in your SASENOAA LIBNAME statement. The NOAA libref is shown inside the DATA step in the SET statement. The SET statement reads observations from the input data set Noaa.cinco and stores them in a SAS data set named Mycinco, as shown in Figure 50.1. 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 3597.

In summary, to specify the NOAA SWDI data set that you want to retrieve, use the NOAASET= option. This required option accepts a string that names the desired NOAA data set, in this case, NOAASET=NX3TVS. The RANGE= option is also required and selects the date range based on the ztime variable, which is the time ID variable for the resulting SAS data set. The Mycinco data set contains the NX3TVS data variables whose observation range is controlled by the RANGE= option. The Mycinco data set contains observations that start May 5, 2006, and end the same day, as specified by the end date May 6, 2006, which is excluded from the selected data. **NOTE:** The begin date on the RANGE= option is inclusive, but the end date is exclusive of the data.

Syntax: SASENOAA Interface Engine

The SASENOAA interface engine uses standard engine syntax to read the observations or data values for NOAA SWDI data sets that can each contain one or more events or time series. Table 50.1 summarizes the options that the SASENOAA engine uses.

Table 50.1 Summary of LIBNAME *libref* SASENOAA Options

Option	Description
AUTOMAP=	Specifies whether or not to overwrite the existing XML map file
BBOX=	Specifies the geographic area to report on by defining a bounding box in the format `'minLon,minLat,maxLon,maxLat'` for minimum longitude, minimum latitude, maximum longitude, maximim latitude. For example: BBOX='-91,30,-90,31'.
CENTER=	Specifies the center point 'longitude,latitude' (to nearest tenth of a degree) of the geographic area to retrieve data for. Use this option with the RADIUS= option to complete the specification.
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 specify CONNECT=ON.
DEBUG=	Specifies whether or not you need diagnostic message logging in the SAS log window
FILTERBY=	Specifies the weather station to retrieve data for
FILTERBYCONDITION=	Specifies the condition for selection of the weather station
FORMAT=	Specifies a file extension that indicates the type of file to retrieve. Only XML, SHP, and KMZ file types are supported for the SASENOAA engine.
ID=	Specifies an ID to retrieve the text message in the warning or the preliminary local storm report databases
KMZMAP=	Specifies the fully qualified filename for the KMZ map that the SASENOAA engine creates. This filename is usually the same as the one in the OUTKMZ= option.
LIMIT=	Specifies the maximum number of observations to use in the report
NOAASET=	Specifies the required NOAA data set name to access in the Severe Weather Data Inventory
OFFSET=	Specifies the offset to the number of observations to start the report
OUTKMZ=	Specifies the name for the downloaded KMZ file. The SASENOAA engine also unzips the KMZ file and gives the KML file this name.
OUTSHP=	Specifies the name for the downloaded SHP ZIP file. The SASENOAA engine also unzips the SHP file and uses this name for the four Esri shapefiles.
OUTXML=	Specifies the name for the XML data that are downloaded from the SWDI web service containing the time series/event data. This name is also used to create the SAS data sets that contain the SWDI data.
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.

Table 50.1 *continued*

Option	Description
RADIUS=	Specifies the radius (in miles, measured from the CENTER= option value's coordinates) of the geographic area to retrieve data for. This option must be used with the CENTER= option.
RANGE=	Specifies the start date and end date for reading the severe weather data in `'YYYYMMDD:YYYYMMDD'` format. The range must be within the same calendar year unless you are requesting statistics only by also specifying the STAT= option. The start date is inclusive of the data, but the end date is exclusive of the data. There is a special option (PERIODOFRECORD) that returns the valid range availability of data for the requested data set specified in the NOAASET= option.
SHPMAP=	Specifies the fully qualified filename for the SHP map that the SASENOAA engine creates. This filename is usually the same as the one in the OUTSHP= option.
STAT=	Specifies the statistical operation that you want to perform on the requested severe weather data
TILE=	Specifies the coordinates of a geographic location to the nearest tenth of a degree. For the earlier example of –95.45,36.88, the matching tile would contain values from –95.4500 to –95.5499 and from 36.8500 to 36.9499.
XMLMAP=	Specifies the fully qualified filename for the XML map that the SASENOAA engine creates. This filename is usually the same as the one in the OUTXML= option.

The LIBNAME *libref* SASENOAA Statement

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

The LIBNAME statement assigns a SAS library reference (libref) to the physical path of the directory where you want the NOAA Severe Weather Data Inventory (SWDI) files to be downloaded and stored. The required *physical-name* argument specifies the location of the folder where your SWDI XML or data shapefiles reside. The *physical-name* should end with a backslash if you are in a Windows environment and a forward slash if you are in a UNIX environment. The designated folder that is specified in the *physical-name* argument must already exist before you submit the LIBNAME *libref* SASENOAA statement.

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

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 SASENOAA 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. The AUTOMAP= option is used only with the XML format (the default).

BBOX='*noaa_bbox_coordinates*'
specifies the coordinates that define the bounding box in the format `'minLon,minLat,maxLon,maxLat'`. This option enables you to select the severe weather data that lie within the geographic area bounded by the box that is defined within the intersections of the specified paired sets of parallels and meridians.

CENTER='*noaa_center_coordinates*'
specifies the center coordinates (longitude, latitude) of a geographic area that, when used along with the RADIUS= option, enable you to select the severe weather data from within the circle whose center is at the specified coordinates (CENTER= option) and of the specified radius (RADIUS= option). An example request follows for "Get all nx3tvs occurring on May 6, 2006, within 15 miles of latitude = 32.7 and longitude = –102.0 and return as XML":

```
LIBNAME libref sasenoaa 'physical-name'
        FORMAT=xml
        NOAASET=nx3tvs
        RANGE='20060506:20060507'
        RADIUS='15.0'
        CENTER='-102.0,32.7'
        OUTXML=mytvs;
```

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

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

FILTERBY='*noaa_filterby_column_value_pair*'
specifies the column name and column value, separated by a colon, to filter the data by. Most often, the column name is WSR_ID and the column value is one of the NEXRAD III weather station ICAO codes shown in Table 50.2.

FILTERBYCONDITION='*noaa_filterbyCond_column_cond_pair*'
specifies the column name and condition value, separated by a colon, to filter the data by. Most often, the column name is WSR_ID and the condition is AND | OR. An example request follows:

```
LIBNAME libref sasenoaa 'physical-name'
        FORMAT=xml
        NOAASET=nx3hail
        RANGE='20110521:20110522'
        FILTERBY='WSR_ID:KFWS'
        FILTERBYCONDITION='WSR_ID:or'
        OUTXML=byNexR;
```

See also the FILTERBY= option.

FORMAT=XML | KMZ | SHP
specifies the format of the file to be retrieved from the NOAA SWDI web service. Although this service can report data in many formats, the SASENOAA engine supports only the XML, SHP, and KMZ formats. When you specify FORMAT=XML, the downloaded data file is named by the OUTXML= option and mapped using the fully designated physical filename specified in the XMLMAP= option. Similarly, when you specify FORMAT=KMZ, use the OUTKMZ= and KMZMAP= options to name your results; and when you specify FORMAT=SHP, use the OUTSHP= and SHPMAP= options to name your results. **NOTE:** Only one format specification is allowed in each SASENOAA LIBNAME statement.

ID='*noaa_id_messageno*'
specifies the message number to retrieve the complete text of the message for the data set specified in the NOAASET= option. ID numbers can be read from the ID column in the results data set (named in the OUTXML= option) for either the warn or plsr data set. The ID= option is used with either the warn or plsr data set to retrieve the entire message that matches the message number indicated in the ID= option for the desired data set, either the NOAA severe storm warnings (warn data set) or the preliminary local storm reports (plsr data set). See Example 50.1 for sample code that shows that the output from the ID= option is placed in the SAS data set named by appending _M to the member name specified in the OUTXML= option.

KMZMAP=*noaa_kmzmapfile*
specifies the fully qualified name of the location where the KMZ map file (zipped KML map file) is automatically stored.

LIMIT=*noaa_limit*
limits the number of observations in the results data set. Specify a number from 1 to 10,000,000.

MAPREF=*noaa_xmlmapref*
specifies the fileref to use for the map assignment. For an example of the SASENOAA engine that uses the MAPREF= and XMLMAP= options in the FILENAME statement to assign a filename, as in the following, see the section "Getting Started: SASENOAA Interface Engine" on page 3582:

```
FILENAME MyMap "U:\noaa950\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 folder that is designated by *physical-name* in your SASENOAA LIBNAME statement, and you can reference it by using the myLib libref. This is shown in the section "Getting Started: SASENOAA Interface Engine" on page 3582, 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 ShearV.

NOAASET=*noaa_data set_SWDI_dsname*
specifies the name of the NOAA SWDI data set that you want to access. Use one of the following names: nx3tvs, nx3meso, nx3mda, nx3hail, nx3structure, plsr, or warn. For a complete description of each data set, see the section "Details: SASENOAA Interface Engine" on page 3592.

OFFSET=*noaa_offset*
specifies a starting row number (offset) in the results to use as your first observation in the results data set.

OUTKMZ=*noaa_kmzfile*
specifies the name of the file where the KMZ data (FORMAT=KMZ) that are returned from the SWDI web service are stored. It is recommended that you specify the OUTKMZ= option when the FORMAT=KMZ option is specified. In cases where the two options do not correspond, the FORMAT= option overrides the designated OUTKMZ= option.

NOTE: The KMZ format produces a ZIP file whose name contains the corresponding file extension (.kmz). The SASENOAA engine automatically unzips the KMZ file to produce a KML map file. The KMZMAP= option gives the name and location of the resulting .kml file.

OUTSHP=*noaa_shpfile*
specifies the name of the file where the SHP data (FORMAT=SHP) that are returned from the SWDI web service are stored. It is recommended that you specify the OUTSHP= option when the FORMAT=SHP option is specified. In cases where the two options do not correspond, the FORMAT= option overrides the designated OUTSHP= option.

NOTE: The SHP format produces a ZIP file whose name contains the corresponding file extension (.shp). The SASENOAA engine automatically unzips the SHP file to produce four Esri map files with the file extensions .dbf, .prj, .shp, and .shx. For example, if OUTSHP=MYSBY and FORMAT=SHP, then the files that contain the SWDI data are named MYSBY.dbf, MYSBY.prj, MYSBY.shp, and MYSBY.shx.

OUTXML=*noaa_xmlfile*
specifies the name of the file where the XML data (FORMAT=XML), KMZ data (FORMAT=KMZ), or SHP data (FORMAT=SHP) that are returned from the SWDI web service are stored. When FORMAT=XML, additional SAS data sets are provided by the SASENOAA engine, depending on two options: ID= and STAT=. When an ID= option is also specified, the engine appends _M to the OUTXML= specification to name the resulting SAS data set that contains the message text that the SWDI web service returns. When the STAT= option is also specified, the engine appends _S to the OUTXML= specification to name the resulting data set that contains the counts from the statistical operation that is performed.

It is recommended that you specify the OUTXML= option when the FORMAT=XML option is specified. In cases where the two options do not correspond, the FORMAT= option overrides the designated OUTXML= option.

PROXY="*noaa_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 *noaa_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.

RADIUS='*noaa_radius***'**
specifies the search radius (in miles) of the area to retrieve the severe weather data for. The current limit for the search radius is 15 miles. This option must be used with the CENTER= option.

RANGE='*noaa_range*'
specifies the date range to report severe (past) weather for. The format for *noaa_range* is `'YYYYMMDD:YYYYMMDD'`. The range must fall within the period of record for the desired data set. The NOAA SWDI data web service returns the period of record for the requested data set (in this case, nx3hail) at the following URL:

http://www.ncdc.noaa.gov/swdiws/xml/nx3hail/periodOfRecord

It also returns a begin date and end date, giving the available time range of data to choose from. Although the limit for a range is one year, often only a few days of data are requested, unless the STAT= option is used. More than one year is allowed in the RANGE= option when you also use the STAT= option to request the COUNT, which returns only the number of observations in the results data set.

SHPMAP=*noaa_shpmapfile*
specifies the fully qualified name of the location where the SHP map file (zipped Esri shapefiles) is automatically stored.

STAT='*noaa_stat_op*'
specifies the statistical operation that you want to perform on the requested severe weather data. You can specify one of the following values for *noaa_stat_op* within single quotes:

COUNT	returns number of results only (no actual data).
COUNTGROUPBY:WSR_ID	returns number of results for each BY group (each WSR_ID that returns data).
TILESUM:*longitude*,*latitude*	returns daily feature counts for a tenth-of-a-degree grid centered at the specified coordinates.

Although the SASENOAA engine automatically checks the statistics to make sure there is a nonzero observation count before requesting the specified data, it is often useful to use the STAT= option to determine the best geographic area and the best date range to retrieve severe weather data that are of the most interest. Output from the STAT= option is placed in the SAS data set named by appending _S to the member name specified in the OUTXML= option.

TILE='*noaa_tile_coordinates*'
specifies that you want to search for severe weather data in the geographic area within a 0.1 degree tile that is centered at the specified coordinates (longitude, latitude).

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

Details: SASENOAA Interface Engine

The SASENOAA interface engine enables SAS programmers to access the NOAA Severe Weather Data Inventory (SWDI) data sets. All dates and times are in Greenwich mean time (GMT), and all latitude and longitude values for input parameters and output data are in the World Geodetic System 1984 (WGS84) datum, the standard for geospatial information.

NOAA Severe Weather Data Inventory Data Sets

The following data sets are supported:

NX3TVS	NEXRAD level III tornado vortex signatures
NX3MESO	NEXRAD level III mesocyclone signatures
NX3MDA	NEXRAD level III digital mesocyclone detection algorithm
NX3HAIL	NEXRAD level III hail signatures
NX3STRUCTURE	NEXRAD level III storm cell structure information
PLSR	Preliminary local storm reports
WARN	Severe thunderstorm, tornado, flash flood, and special marine warnings

To display details about the available inventory for the NEXRAD level III data sets, enter the following URL in your browser:

http://www.ncdc.noaa.gov/swdiws/xml

The result is a list of available SWDI web service data sets, each with a description, begin date, end date, tile summary allowed (yes or no), and ID query allowed (yes or no). The web page at the following URL describes the column definitions and units for each NEXRAD III product and includes a discussion about accuracy:

http://www.ncdc.noaa.gov/swdiws/csv/nx3hail:inv

NOAA NEXRAD Sites and Their ICAO Codes and Coordinates

A list of the NEXRAD sites and their corresponding WSR_ID codes, also known as International Civil Aviation Organization (ICAO) codes, is given in Table 50.2. For examples of how to use this important BY variable, WSR_ID, to subset and gather statistics about NOAA SWDI data, see Example 50.6 and Example 50.4.

Table 50.2 List of NEXRAD Sites and Their Coordinates

State	City	ICAO Location Identifier	Coordinates
PR	San Juan	TJUA	18.1155998°N 66.0780644°W
ME	Loring AFB	KCBW	46.0391944°N 67.8066033°W
ME	Portland	KGYX	43.8913555°N 70.2565545°W
VT	Burlington	KCXX	44.5109941°N 73.166424°W
MA	Boston	KBOX	41.9558919°N 71.1369681°W
NY	Albany	KENX	42.5865699°N 74.0639877°W
NY	Binghamton	KBGM	42.1997045°N 75.9847015°W
NY	Buffalo	KBUF	42.9488055°N 78.7369108°W
NY	Montague	KTYX	43.7556319°N 75.6799918°W
NY	New York City	KOKX	40.8655093°N 72.8638548°W
DE	Dover AFB	KDOX	38.8257651°N 75.4400763°W
PA	Philadelphia	KDIX	39.9470885°N 74.4108027°W
PA	Pittsburgh	KPBZ	40.5316842°N 80.2179515°W
PA	State College	KCCX	40.9228521°N 78.0038738°W
WV	Charleston	KRLX	38.3110763°N 81.7229015°W
VA	Norfolk/Richmond	KAKQ	36.9840475°N 77.007342°W
VA	Roanoke	KFCX	37.0242098°N 80.2736664°W
VA	Sterling	KLWX	38.9753957°N 77.4778444°W
NC	Morehead City	KMHX	34.7759313°N 76.8762571°W
NC	Raleigh/Durham	KRAX	35.6654967°N 78.4897855°W
NC	Wilmington	KLTX	33.9891631°N 78.4291059°W
SC	Charleston	KCLX	32.6554866°N 81.0423124°W
SC	Columbia	KCAE	33.9487579°N 81.1184281°W
SC	Greer	KGSP	34.8833435°N 82.2200757°W
GA	Atlanta	KFFC	33.3635771°N 84.565866°W
GA	Moody AFB	KVAX	30.8903853°N 83.0019021°W
GA	Robins AFB	KJGX	32.6755239°N 83.3508575°W
FL	Eglin AFB	KEVX	30.5649908°N 85.921559°W
FL	Jacksonville	KJAX	30.4846878°N 81.7018917°W
FL	Key West	KBYX	24.5974996°N 81.7032355°W
FL	Melbourne	KMLB	28.1131808°N 80.6540988°W
FL	Miami	KAMX	25.6111275°N 80.412747°W
FL	Tallahassee	KTLH	30.397568°N 84.3289116°W
FL	Tampa	KTBW	27.7054701°N 82.40179°W
AL	Birmingham	KBMX	33.1722806°N 86.7698425°W
AL	Fort Rucker	KEOX	31.4605622°N 85.4592401°W
AL	Huntsville	KHTX	34.930508°N 86.0837388°W
AL	Maxwell AFB	KMXX	32.5366608°N 85.7897848°W
AL	Mobile	KMOB	30.6795378°N 88.2397816°W
MS	Brandon/Jackson	KDGX	32.2797358°N 89.9846309°W
MS	Columbus AFB	KGWX	33.8967796°N 88.3293915°W
TN	Knoxville/ Tri Cities	KMRX	36.168538°N 83.401779°W
TN	Memphis	KNQA	35.3447802°N 89.8734534°W
TN	Nashville	KOHX	36.2472389°N 86.5625185°W

Table 50.2 *continued*

State	City	ICAO Location Identifier	Coordinates
KY	Fort Campbell	KHPX	36.7368894°N 87.2854328°W
KY	Jackson	KJKL	37.590762°N 83.313039°W
KY	Louisville	KLVX	37.9753058°N 85.9438455°W
KY	Paducah	KPAH	37.0683618°N 88.7720257°W
OH	Cleveland	KCLE	41.4131875°N 81.8597451°W
OH	Wilmington	KILN	39.5083314°N 83.8176925°W
MI	Detroit/Pontiac	KDTX	42.6999677°N 83.471809°W
MI	Gaylord	KAPX	44.907106°N 84.719817°W
MI	Grand Rapids	KGRR	42.893872°N 85.5449206°W
MI	Marquette	KMQT	46.5311443°N 87.5487131°W
IN	Evansville	KVWX	38.2603901°N 87.7246553°W
IN	Indianapolis	KIND	39.7074962°N 86.2803675°W
IN	North Webster	KIWX	41.3586356°N 85.7000488°W
IL	Chicago	KLOT	41.6044264°N 88.084361°W
IL	Lincoln	KILX	40.150544°N 89.336842°W
WI	Green Bay	KGRB	44.4984644°N 88.111124°W
WI	La Crosse	KARX	43.822766°N 91.1915767°W
WI	Milwaukee	KMKX	42.9678286°N 88.5506335°W
MN	Duluth	KDLH	46.8368569°N 92.2097433°W
MN	Minneapolis/ St. Paul	KMPX	44.8488029°N 93.5654873°W
IA	Davenport	KDVN	41.611556°N 90.5809987°W
IA	Des Moines	KDMX	41.7311788°N 93.7229235°W
MO	Kansas City	KEAX	38.8102231°N 94.2644924°W
MO	Springfield	KSGF	37.235223°N 93.4006011°W
MO	St. Louis	KLSX	38.6986863°N 90.682877°W
AR	Fort Smith	KSRX	35.2904423°N 94.3619075°W
AR	Little Rock	KLZK	34.8365261°N 92.2621697°W
LA	Fort Polk	KPOE	31.1556923°N 92.9762596°W
LA	Lake Charles	KLCH	30.125382°N 93.2161188°W
LA	New Orleans	KLIX	30.3367133°N 89.8256618°W
LA	Shreveport	KSHV	32.450813°N 93.8412774°W
TX	Amarillo	KAMA	35.2334827°N 101.7092478°W
TX	Austin/ San Antonio	KEWX	29.7039802°N 98.028506°W
TX	Brownsville	KBRO	25.9159979°N 97.4189526°W
TX	Corpus Christi	KCRP	27.7840203°N 97.511234°W
TX	Dallas/Ft. Worth	KFWS	32.5730186°N 97.3031911°W
TX	Dyess AFB	KDYX	32.5386009°N 99.2542863°W
TX	El Paso	KEPZ	31.8731115°N 106.697942°W
TX	Fort Hood	KGRK	30.7217637°N 97.3829627°W
TX	Houston/ Galveston	KHGX	29.4718835°N 95.0788593°W

Table 50.2 *continued*

State	City	ICAO Location Identifier	Coordinates
TX	Laughlin AFB	KDFX	29.2730823°N 100.2802312°W
TX	Lubbock	KLBB	33.6541242°N 101.814149°W
TX	Midland/ Odessa	KMAF	31.9433953°N 102.1894383°W
TX	San Angelo	KSJT	31.3712815°N 100.4925227°W
OK	Frederick	KFDR	34.3620014°N 98.9766884°W
OK	Oklahoma City	KTLX	35.3333873°N 97.2778255°W
OK	Tulsa	KINX	36.1750977°N 95.5642802°W
OK	Vance AFB	KVNX	36.7406166°N 98.1279409°W
KS	Dodge City	KDDC	37.7608043°N 99.9688053°W
KS	Goodland	KGLD	39.3667737°N 101.7004341°W
KS	Topeka	KTWX	38.996998°N 96.232618°W
KS	Wichita	KICT	37.6545724°N 97.4431461°W
NE	Grand Island/ Hastings	KUEX	40.320966°N 98.4418559°W
NE	North Platte	KLNX	41.9579623°N 100.5759609°W
NE	Omaha	KOAX	41.3202803°N 96.3667971°W
SD	Aberdeen	KABR	45.4558185°N 98.4132046°W
SD	Rapid City	KUDX	44.1248485°N 102.8298157°W
SD	Sioux Falls	KFSD	43.5877467°N 96.7293674°W
ND	Bismarck	KBIS	46.7709329°N 100.7605532°W
ND	Grand Forks (Mayville)	KMVX	47.5279417°N 97.3256654°W
ND	Minot AFB	KMBX	48.39303°N 100.8644378°W
MT	Billings	KBLX	45.8537632°N 108.6068165°W
MT	Glasgow	KGGW	48.2064536°N 106.6252971°W
MT	Great Falls	KTFX	47.4595023°N 111.3855368°W
MT	Missoula	KMSX	47.0412971°N 113.9864373°W
WY	Cheyenne	KCYS	41.1519308°N 104.8060325°W
WY	Riverton	KRIW	43.0660779°N 108.4773731°W
CO	Denver	KFTG	39.7866156°N 104.5458126°W
CO	Grand Junction	KGJX	39.0619824°N 108.2137012°W
CO	Pueblo	KPUX	38.4595034°N 104.1816223°W
NM	Albuquerque	KABX	35.1497579°N 106.8239576°W
NM	Cannon AFB	KFDX	34.6341569°N 103.6186427°W
NM	Holloman AFB	KHDX	33.0768844°N 106.1200923°W
AZ	Flagstaff	KFSX	34.574449°N 111.198367°W
AZ	Phoenix	KIWA	33.289111°N 111.6700092°W
AZ	Tucson	KEMX	31.8937186°N 110.6304306°W
AZ	Yuma	KYUX	32.4953477°N 114.6567214°W
UT	Cedar City	KICX	37.5931771°N 112.8637719°W
UT	Salt Lake City	KMTX	41.2627795°N 112.4480081°W
ID	Boise	KCBX	43.4902104°N 116.2360436°W

Table 50.2 *continued*

State	City	ICAO Location Identifier	Coordinates
ID	Pocatello/ Idaho Falls	KSFX	43.1055967°N 112.6860487°W
NV	Elko	KLRX	40.7396933°N 116.8025529°W
NV	Las Vegas	KESX	35.7012894°N 114.8918277°W
NV	Reno	KRGX	39.7541931°N 119.4620597°W
CA	Beale AFB	KBBX	39.4956958°N 121.6316557°W
CA	Edwards AFB	KEYX	35.0979358°N 117.5608832°W
CA	Eureka	KBHX	40.4986955°N 124.2918867°W
CA	Los Angeles	KVTX	34.4116386°N 119.1795641°W
CA	Sacramento	KDAX	38.5011529°N 121.6778487°W
CA	San Diego	KNKX	32.9189891°N 117.041814°W
CA	San Francisco	KMUX	37.155152°N 121.8984577°W
CA	San Joaquin Valley	KHNX	36.3142088°N 119.6320903°W
CA	Santa Ana Mountains	KSOX	33.8176452°N 117.6359743°W
CA	Vandenberg AFB	KVBX	34.8383137°N 120.3977805°W
HI	Kauai	PHKI	21.8938762°N 159.5524585°W
HI	Kohala	PHKM	20.1254606°N 155.778054°W
HI	Molokai	PHMO	21.1327531°N 157.1802807°W
HI	South Shore	PHWA	19.0950155°N 155.5688846°W
OR	Medford	KMAX	42.0810766°N 122.7173334°W
OR	Pendleton	KPDT	45.6906118°N 118.8529301°W
OR	Portland	KRTX	45.7150308°N 122.9650542°W
WA	Langley Hill	KLGX	47.116806°N 124.10625°W
WA	Seattle/Tacoma	KATX	48.1945614°N 122.4957508°W
WA	Spokane	KOTX	47.6803744°N 117.6267797°W
AK	Bethel	PABC	60.791987°N 161.876539°W
AK	Fairbanks/ Pedro Dome	PAPD	65.0351238°N 147.5014222°W
AK	Kenai	PAHG	60.6156335°N 151.2832296°W
AK	King Salmon	PAKC	58.6794558°N 156.6293335°W
AK	Middleton Island	PAIH	59.46194°N 146.30111°W
AK	Nome	PAEC	64.5114973°N 165.2949071°W
AK	Sitka/ Biorka Island	PACG	56.85214°N 135.552417°W
GU	Andersen AFB	PGUA	13.455965°N 144.8111022°E
NA	Lajes Field, Azores	LPLA	38.73028°N 27.32167°W
SK	Camp Humphreys, South Korea	RKSG	37.207652°N 127.285614°E
SK	Kunsan Air Base, South Korea	RKJK	35.92417°N 126.62222°E
JP	Kadena Air Base, Japan	RODN	26.30194°N 127.90972°E

SAS Output Data Set

You can use a SAS DATA step to write the selected NOAA Severe Weather Data Inventory data to a SAS data set. This enables you to use SAS software to easily perform data analysis. 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 sets are described in the section "Examples: SASENOAA Interface Engine" on page 3600 and summarized in Table 50.3 through Table 50.7. Each type of SWDI data set contains its own columns and variables, and the resulting SAS data set is named by the OUTXML= option specification. When the ID= option is used, another SAS data set is created with _M appended to the original data set name, and if the STAT= option is used, then another data set is created with _S appended to the original data set 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 SASENOAA engine libref to create a custom view of your data.

Table 50.3 NX3HAIL NEXRAD Level III Hail Data Set

Variable Name	Description
wsr_id	NEXRAD or Terminal Doppler Weather Radar (TDWR) site ID
cell_id	Cell ID unique to radar site
prob	Probability of hail (percentage)
sevprob	Probability of severe hail (percentage)
maxsize	Maximum size of hail (in)

Table 50.4 NX3MESO NEXRAD Level III Legacy Mesocyclone Data Set

Variable Name	Description
wsr_id	NEXRAD or Terminal Doppler Weather Radar (TDWR) site ID
cell_id	Cell ID unique to radar site
cell_type	'Meso', '3dc shr', or 'unc shr'
range	Range (naut. miles)
azimuth	Azimuth (deg)
base_height	Base height of feature (kft)
height	Height of feature (kft)
radial_diam	Diameter of feature along range axis (naut. mi)
az_diam	Diameter of feature in azimuth angle (deg)
shear	Wind shear (E-3/s)

Table 50.5 NX3STRUCTURE NEXRAD Level III Storm Structure Data Set

Variable Name	Description
wsr_id	NEXRAD or Terminal Doppler Weather Radar (TDWR) site ID
cell_id	Cell ID unique to radar site
range	Range (naut. mi)
azimuth	Azimuth (deg)
vil	Vertically integrated liquid (kg/m^2)
max_reflect	Maximum reflectivity (dbz)

Table 50.6 NX3TVS NEXRAD Level III Tornado Vortex Signature Data Set

Variable Name	Description
wsr_id	NEXRAD or Terminal Doppler Weather Radar (TDWR) site ID
cell_id	Cell ID unique to radar site
range	Range (naut. mi)
azimuth	Azimuth (deg)
max_shear	Maximum shear (E-3/s)
mxdv	Maximum delta-velocity (knots)

Table 50.7 NX3MDA NEXRAD Level III Digital Mesocyclone Detection Algorithm Data Set

Variable Name	Description
wsr_id	NEXRAD or Terminal Doppler Weather Radar (TDWR) site ID
cell_id	Cell ID unique to radar site
str_rank	Strength ranking
scit_id	ID in storm cell identification and tracking (SCIT) algorithm
range	Range (naut. mi)
azimuth	Azimuth (deg)
ll_rot_vel	Low-level rotational velocity (kt)
ll_dv	Low-level delta-velocity (kt)
ll_base	Base (kft)
depth_kft	Depth (kft)
dpth_stmrl	Storm-relative depth (percentage)
max_rv_kft	Maximum rotational velocity height (kft)
max_rv_kts	Maximum rotational velocity (knots)
tvs	Tornado vortex signature (yes or no)
motion_deg	Motion direction (deg)
motion_kts	Motion speed (kts)
msi	Mesocyclone strength index

The storm cell identification and tracking (SCIT) algorithm is an enhanced WSR-88D algorithm that is outside the scope of this chapter, but this section briefly summarizes some of the variables in the NX3MDA

data set. Storm-relative depth is the ratio (expressed in percentage) of meso-depth divided by the storm depth as determined by the SCIT algorithm's cell. Strength ranking and mesocyclone strength index (MSI) are nondimensional numbers that provide a way to determine the 3D-integrated intensity value of the detection.

Max_rv_kft is the height (in kilofeet) at which maximum rotational velocity was detected; it might or might not be associated with the lowest radar elevation angle. Max_rv_kts is the rotational velocity in knots; it might or might not be associated with the lowest radar elevation angle. The variables ll_rot_vel, ll_dv, and ll_base are always associated with the lowest elevation angle, so max_ and ll_ values are sometimes identical.

SAS OUTXML File

The SAS XML (XML format) data that are returned by the NOAA SWDI web service are placed in a file that is named by the OUTXML= option. The SASENOAA interface engine creates a separate XML file for each SAS data set that is created. By default, OUTXML=NOAA, which creates a file named NOAA.xml in the current working directory. The SAS data set created when the XML data are read into SAS is placed in the folder specified by the physical path in the LIBNAME libref SASENOAA statement, which is described in the section "The LIBNAME *libref* SASENOAA Statement" on page 3587. The name that you specify in the OUTXML= option is also used to form the names of other data sets, but a suffix is added to the name to maintain the identity of the file, such as _M for the message file data set (ID= option) and _S for the statistics results data set (STAT= option).

SAS XML Map File

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

Virtual Globe Mapping Output and ZIP Files

When you specify the FORMAT=KMZ option, the SASENOAA interface engine requests the SWDI data in KMZ format. This results in the retrieval of a zipped KML file, which is then unzipped, saved with the .kml extension, and named by the OUTKMZ= option. In addition, the corresponding KMZ file is saved in the location specified by the fully qualified filename given in the KMZMAP= option. You can then use virtual globe software provided by Google Maps to import your KML data so that you can visualize the results both geospatially and timewise by holding the mouse pointer over each data point to see the variable values that correspond to the requested NOAA data set.

When you specify the FORMAT=SHP option, the SASENOAA engine requests the SWDI data in SHP format. This results in the retrieval of a zipped SHP file, which is then unzipped; the four resulting files are saved with the extensions .dbf, .prj, .shp, and .shx and named by the OUTSHP= option. In addition, the corresponding SHP ZIP file is saved in the location that is specified by the fully qualified filename given in the SHPMAP= option. You can then use virtual globe software such as SAS Bridge for Esri or use PROC MAPIMPORT and PROC GMAP to map your results.

SAS KMZ Map File

The KMZ map (by default) is automatically created and placed in the file that is named by the fully qualified filename specified in the KMZMAP= option of the LIBNAME *libref* SASENOAA statement. The SASENOAA interface engine invokes PROC HTTP to create the map and to read the KMZ data into SAS.

SAS OUTKMZ File

The SAS KMZ (zipped KML format) data that are returned by the NOAA SWDI web service are placed in a file that is named by the OUTKMZ= option. The SASENOAA interface engine unzips the KMZ file and creates a separate KML file for each SASENOAA engine libref. The SAS KML data file is given the name specified by the OUTKMZ= option and is placed in the location that is specified by the *physical-name* in your LIBNAME *libref* SASENOAA statement, which is described in the section "The LIBNAME *libref* SASENOAA Statement" on page 3587.

SAS OUTSHP File

The SAS SHP (zipped Esri shapefiles format) data that are returned by the NOAA SWDI web service are placed in a file that is named by the OUTSHP= option. The SASENOAA interface engine creates a separate SHP ZIP file for each SASENOAA engine libref. The SASENOAA engine unzips the SHP data file, creating four files that are given the name specified by the OUTSHP= option plus the four file extensions (.dbf, .prj, .shp, and .shx). The four files are saved in the location that is specified by the *physical-name* in your LIBNAME *libref* SASENOAA statement, which is described in the section "The LIBNAME *libref* SASENOAA Statement" on page 3587.

SAS SHP Map File

The SHP map (by default) is automatically created and placed in the file that is named by the fully qualified filename specified in the SHPMAP= option of the LIBNAME *libref* SASENOAA statement. The SASENOAA interface engine invokes PROC HTTP to create the map and to read the SHP data into SAS.

Examples: SASENOAA Interface Engine

Example 50.1: Retrieving Severe Storm Warning Data with ID= Option for a Specific Date

This example shows how to use the RANGE= option to retrieve severe storm warning data for a specific date. It also shows how to use the ID= option to read the message text for one ID (ID='397190'). When the ID= option is used, there are two output data sets. The first data set consists of the warning results data (named C1nco in the OUTXML= option), which contain the actual list of storm warnings for the date range that is specified in the RANGE= option. The second data set, C1nco_M, contains the text of the message ID specified in the ID= option, which in this example is 397190.

The output of the PRINT procedure for the Myc1nco data set is shown in Output 50.1.1.

```
options validvarname=any;

title 'Retrieve Warning Data with ID= Option for May 5, 2006';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=warn
   id='397190'                /* create c1nco_m data set */
   range='20060505:20060506'
   outXml=c1nco               /* create c1nco  data set */
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\c1nco.map"
   format=xml;

data mylib.myc1nco;
   set noaa.c1nco;
run;

proc contents data=mylib.myc1nco; run;
proc print data=mylib.myc1nco(obs=5); run;
```

Output 50.1.1 NOAA Severe Storm Warnings with ID= Option for May 5, 2006

Retrieve Warning Data with ID= Option for May 5, 2006

Obs	ztime_start	ztime_end	id	warningtype	issuewfo	messageid
1	2006-05-04T11:57:00	2006-05-05T05:45:00	397088	FLASH FLOOD	KSGF	41157
2	2006-05-04T22:50:00	2006-05-05T00:15:00	397156	SPECIAL MARINE	KLIX	42251
3	2006-05-04T22:50:00	2006-05-05T00:15:00	397157	SPECIAL MARINE	KLIX	42251
4	2006-05-04T23:07:00	2006-05-05T00:00:00	397161	SEVERE THUNDERSTORM	KSHV	42307
5	2006-05-04T23:10:00	2006-05-05T00:00:00	397162	SEVERE THUNDERSTORM	KJAN	42310

Obs	shape
1	POLYGON ((-95.02 37.64, -95.02 37.02, -94.57 37.03, -94.59 36.52, -94.1 36.51, -94.12 37.62, -95.02 37.64))
2	POLYGON ((-90.06 29.34, -89.8 29.15, -89.55 29.26, -89.61 29.27, -89.6 29.35, -89.67 29.31, -89.77 29.33, -89.75 29.41, -89.81 29.43, -89.83 29.49, -89.93 29.51, -89.94 29.48, -90.07 29.55, -90.17 29.51, -90.06 29.43, -90.06 29.34))
3	POLYGON ((-90.06 29.34, -89.8 29.15, -89.55 29.26, -89.61 29.27, -89.6 29.35, -89.67 29.31, -89.77 29.33, -89.75 29.41, -89.81 29.43, -89.83 29.49, -89.93 29.51, -89.94 29.48, -90.07 29.55, -90.17 29.51, -90.06 29.43, -90.06 29.34))
4	POLYGON ((-94.09 31.63, -94.04 31.6, -93.95 31.6, -93.93 31.61, -93.84 31.6, -93.8 31.71, -93.84 31.78, -94.11 31.78, -94.13 31.63, -94.09 31.63))
5	POLYGON ((-91.57 33.33, -91.73 33.01, -91.17 33.02, -91.14 33.07, -91.2 33.14, -91.09 33.16, -91.14 33.29, -91.57 33.33))

The data sets, C1nco and C1nco_M, reside in the test folder, because that is the physical path given in the SASENOAA LIBNAME statement inside the double quotes:

```
libname noaa sasenoaa "U:\noaa940\test\"
```

NOTE: The DATA step that creates the Mylib.Myc1nco data set reads only the C1nco data into the document folder that is specified by the Mylib libref:

```
libname mylib "U:\noaa940\doc\";
```

But the other data set, which contains the message text data set, C1nco_M, is not copied into the document folder; instead it remains in the test folder where it was originally created by the SASENOAA engine. You could also copy it into the document folder using the following code:

```
libname myMes "U:\noaa940\test\";

data mylib.myc1nco_M;
   set myMes.c1nco_M;
run;
```

You should not use the SASENOAA engine libref (NOAA) to access the already created SAS data set C1nco_M, because the message results were already placed in that data set automatically when you ran the example code to download the XML from the SWDI web service. The ID= option causes the SASENOAA engine to create the second data set, C1nco_M. After you read the data into SAS, you should use the normal Base SAS engine to access the resulting SAS data sets, by using the myMes libref in the SET statement that invokes the Base SAS engine.

Example 50.2: Retrieving a Preliminary Local Storm Report by Using a Bounding Box

This example shows how to use a bounding box (by specifying the BBOX= option) to define the geographic area to retrieve a preliminary local storm report (PLSR) starting May 5 and ending May 10 (not including May 10). The output is shown in Output 50.2.1 for the data set My8bb and in Output 50.2.2 for the data set My8bb_M.

```
options validvarname=any;

title 'Retrieve the NOAA SWDI PLSR Data for a Bounding Box';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=plsr
   range='20060505:20060510'
   bbox='-91,30,-90,31'
   id='427200'
   outXml=my8BB
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\my8BB.map"
   format=xml
   ;
```

```
data mylib.PLSRbb;
   set noaa.my8BB;
run;

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

Output 50.2.1 Preliminary Local Storm Report for a Bounding Box with the RANGE= Option

Retrieve the NOAA SWDI PLSR Data for a Bounding Box

Obs	ztime	id	event	magnitude	city	county	state	source	shape
1	2006-05-09T02:20:00	427540	HAIL	1	5 E KENTWOOD	TANGIPAHOA	LA	TRAINED SPOTTER	POINT (-90.43 30.93)
2	2006-05-09T02:40:00	427536	HAIL	1	MOUNT HERMAN	WASHINGTON	LA	TRAINED SPOTTER	POINT (-90.3 30.96)
3	2006-05-09T02:40:00	427537	TSTM WND DMG	-9999	MOUNT HERMAN	WASHINGTON	LA	TRAINED SPOTTER	POINT (-90.3 30.96)
4	2006-05-09T03:00:00	427199	HAIL	0	FRANKLINTON	WASHINGTON	LA	AMATEUR RADIO	POINT (-90.14 30.85)
5	2006-05-09T03:17:00	427200	TORNADO	-9999	5 S FRANKLINTON	WASHINGTON	LA	LAW ENFORCEMENT	POINT (-90.14 30.78)

The RANGE= option selects only the storm reports for dates from May 5 to May 10, 2006 (not including May 10), and the BBOX= option limits the data returned to the geographic area defined by the intersection of the specified coordinates: minimum longitude, minimum latitude, maximum longitude, and maximum latitude. The ID='427200' option returns additional data in the SAS data set my8bb_M for the storm event that has that ID, and the results can be viewed using the following sample code. **NOTE:** The SASENOAA engine appends _M to the name specified in the OUTXML= option for these additional data.

```
libname myreport "U:\noaa940\test\";

proc contents data=myreport.my8bb_m; run;
proc print data=myreport.my8bb_m; run;
```

Output 50.2.2 Preliminary Local Storm Report for Tornado Event, ID=427200

Retrieve the NOAA SWDI PLSR Data for a Bounding Box

Obs	remarks
1	TORNADO MOVED ACROSS HWY 25 BLEW TWO CARS IN THE DITCH AND DEBRIS ON HWY.

Example 50.3: Retrieving Mesocyclone Data for a Specific Date

This example shows how to retrieve mesocyclone data for a specific date. The NX3MESO legacy database displays information about the existence and nature of rotations associated with thunderstorms. Numerical output includes the azimuth, range, and height of the mesocyclone. Output 50.3.1 shows the NX3MESO data for RANGE='20060505:20060506'. **NOTE:** The end date, May 6, 2006, is exclusive of the data.

```
title 'Mesocyclone Data for May 5, 2006';
libname _all_ clear;
options validvarname=any;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=nx3meso
   range='20060505:20060506'   /*  stat='countGroupBy:WSR_ID' */
   outxml=c3nco
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\c3nco.map"
   format=xml
   ;

data mylib.myc3nco;
   set noaa.c3nco;
run;

proc contents data=mylib.myc3nco; run;
proc print data=mylib.myc3nco(obs=10); run;
```

Output 50.3.1 Mesocyclone Data for May 5, 2006

Mesocyclone Data for May 5, 2006

Obs	ztime	wsr_id	cell_id	cell_type	range	azimuth	base_height
1	2006-05-05T00:00:45	KLBB	S0	UNC SHR	73	223	13.5
2	2006-05-05T00:00:45	KLBB	G0	UNC SHR	53	226	15.0
3	2006-05-05T00:00:45	KLBB	P0	MESO	122	165	16.5
4	2006-05-05T00:00:54	KFWS	R4	UNC SHR	59	224	17.1
5	2006-05-05T00:00:59	KDYX	A2	UNC SHR	114	247	16.3
6	2006-05-05T00:00:59	KDYX	Y0	UNC SHR	97	183	23.2
7	2006-05-05T00:01:55	KIND	NULL	UNC SHR	15	125	0.9
8	2006-05-05T00:01:57	KEWX	L1	MESO	93	319	15.5
9	2006-05-05T00:01:57	KEWX	L1	UNC SHR	97	317	16.4
10	2006-05-05T00:01:57	KEWX	L1	UNC SHR	103	316	17.8

Obs	top_height	height	radial_diam	az_diam	shear	shape
1	13.5	13.5	1.8	3.7	8	POINT (-102.798000555293 32.7586300599108)
2	15.0	15.0	2.0	3.7	7	POINT (-102.569931348078 33.0369811650688)
3	21.5	16.5	1.9	4.2	9	POINT (-101.197496803559 31.6843740429353)
4	17.1	17.1	4.0	3.2	32	POINT (-98.1051400587857 31.861696176778)
5	16.3	16.3	2.0	3.4	9	POINT (-101.306015945194 31.7772827698164)
6	23.2	23.2	1.5	6.9	6	POINT (-99.3523385786414 30.9200780684841)
7	0.9	0.9	2.3	1.1	76	POINT (-86.0151526678541 39.5642056660823)
8	24.7	20.0	5.7	7.6	12	POINT (-99.2095322445991 30.8712677231185)
9	16.4	16.4	5.3	7.6	5	POINT (-99.3092294394828 30.8829373004521)
10	17.8	17.8	2.0	2.2	8	POINT (-99.4144574037648 30.9345286761316)

The results are sorted by the ztime variable, along with WSR_ID, which is a BY variable that can be

referenced in the STAT= option (such as STAT='COUNTGROUPBY:WSR_ID') or in the FILTERBY= option (such as FILTERBY='WSR_ID:KBLX'). For a list of possible values for WSR_ID, see the ICAO Location Identifiers column in Table 50.2.

Example 50.4: Retrieving Hail Data for One Weather Station

This example shows how to use the FILTERBY= and FILTERBYCONDITION= options to retrieve the data for one weather station (WSR_ID=KFWS) by using the hail storm data from the NX3HAIL database for May 21, 2011. The output is shown in Output 50.4.1.

```
options validvarname=any;

title 'Retrieve NX3HAIL Data for WSR_ID=KFWS on May 21, 2011';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=nx3hail
   range='20110521:20110522'
   filterBy='WSR_ID:KFWS'
   filterByCondition='WSR_ID:or'
   outXml=myCby
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\myCby.map"
   format=XML
   ;

data mylib.HAILbyC;
   set noaa.myCby;
run;

proc contents data=mylib.HAILbyC; run;
proc print data=mylib.HAILbyC(obs=10); run;
```

Output 50.4.1 Severe Hail Storm Data Using FILTERBY= Option for Weather Station KFWS on May 21, 2011

Retrieve NX3HAIL Data for WSR_ID=KFWS on May 21, 2011

Obs	ztime	wsr_id	cell_id	prob	sevprob	maxsize	shape
1	2011-05-21T00:09:38	KFWS	U8	100	30	0.75	POINT (-96.7920955739634 31.1345064213377)
2	2011-05-21T00:09:38	KFWS	X1	100	50	1.00	POINT (-95.4408845222209 31.6765991602025)
3	2011-05-21T00:09:38	KFWS	E9	100	50	1.00	POINT (-96.5897740989292 31.4208018135191)
4	2011-05-21T00:09:38	KFWS	G6	100	40	1.00	POINT (-96.8664936098099 31.0651213629879)
5	2011-05-21T00:09:38	KFWS	R0	100	40	1.00	POINT (-96.1199975968128 31.4803477097816)
6	2011-05-21T00:09:38	KFWS	I1	100	60	1.25	POINT (-96.9732130383308 30.9606322478281)
7	2011-05-21T00:09:38	KFWS	L8	100	50	1.25	POINT (-96.253873691341 31.5350758385099)
8	2011-05-21T00:09:38	KFWS	R7	100	70	1.50	POINT (-96.5356562569208 31.3319818714777)
9	2011-05-21T00:09:38	KFWS	I3	100	80	1.75	POINT (-96.6051331772435 31.0844364854615)
10	2011-05-21T00:18:07	KFWS	L2	100	20	0.75	POINT (-97.1877409442769 30.6713908258023)

You can see that the output data set, myCby, returns only the data for WSR_ID=‘KFWS’ because of the FILTERBY= and FILTERBYCONDITION= options.

Example 50.5: Retrieving Tornado Vortex Signature Data within a Distance Specified by a Center and a Radius

When you specify NOAASET=NX3TVS, you retrieve data that show an intense gate-to-gate azimuthal shear associated with tornadic-scale rotation. This example shows how to search the NX3TVS database by using the RADIUS= and CENTER= options to retrieve tornado vortex signature data for the date range from May 5 to May 16, 2006. The output is shown in Output 50.5.1.

```
options validvarname=any;

title 'Tornado Vortex Signatures with CENTER= and RADIUS= Options for a Date Range';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=nx3tvs
   range='20060505:20060516'
   radius='15.0'
   center='-102.0,32.7'
   outxml=my2CR
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\my2CR.map"
   format=xml
   ;

data mylib.TVS2CR;
   set noaa.my2CR;
run;

proc contents data=mylib.TVS2CR; run;
proc print data=mylib.TVS2CR(obs=10); run;
```

Output 50.5.1 NX3TVS Data Search Using CENTER= and RADIUS= Options

Tornado Vortex Signatures with CENTER= and RADIUS= Options for a Date Range

Obs	ztime	wsr_id	cell_id	cell_type	range	azimuth	max_shear	mxdv	shape
1	2006-05-05T00:05:50	KBMX	Q0	TVS	7	217	403	116	POINT (-86.8535716274277 33.0786326913943)
2	2006-05-05T00:10:02	KBMX	Q0	TVS	5	208	421	120	POINT (-86.8165772540846 33.0982820681588)
3	2006-05-05T00:12:34	KSJT	P2	TVS	49	106	17	52	POINT (-99.5771091971025 31.1421609654838)
4	2006-05-05T00:17:31	KSJT	B4	TVS	40	297	25	62	POINT (-101.188161700093 31.672392833416)
5	2006-05-05T00:29:13	KMAF	H4	TVS	53	333	34	111	POINT (-102.664426480293 32.7306917937698)
6	2006-05-05T00:31:25	KLBB	N0	TVS	51	241	24	78	POINT (-102.70047613441 33.2380072329615)
7	2006-05-05T00:33:25	KMAF	H4	TVS	52	334	46	145	POINT (-102.6393683028 32.7226656893341)
8	2006-05-05T00:37:37	KMAF	H4	TVS	50	334	34	107	POINT (-102.621904684258 32.6927081076156)
9	2006-05-05T00:41:51	KMAF	H4	TVS	51	335	29	91	POINT (-102.614794815627 32.714139844846)
10	2006-05-05T00:44:33	KLBB	N0	TVS	46	245	35	100	POINT (-102.643380529494 33.3266446067682)

Example 50.6: Retrieving Digital Mesocyclone Detection Algorithm Data for a Specific Date

The digital mesocyclone detection algorithm data (NX3MDA) are the successor to the legacy mesocyclone data (NX3MESO) and are designed to display information about the existence and nature of rotations associated with thunderstorms. Numerical output includes the azimuth, range, and height of the mesocyclone. This example retrieves these data for June 8, 2016. The first 10 observations are shown in Output 50.6.1.

```
options validvarname=any;

title 'Digital Mesocyclone Detection Algorithm Data for June 8, 2016';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=nx3mda
   range='20160608:20160609'
   stat='countGroupBy:WSR_ID'  /* need this to create c9nco_S */
   outXml=c9nco
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\c9nco.map"
   format=xml
   ;

data mylib.myc9nco;
```

```
   set noaa.c9nco;
run;

proc contents data=mylib.myc9nco; run;
proc print data=mylib.myc9nco(obs=10); run;
```

Output 50.6.1 Digital Mesocyclone Detection Algorithm Data for June 8, 2016

Digital Mesocyclone Detection Algorithm Data for June 8, 2016

Obs	ztime	wsr_id	cell_id	str_rank	scit_id	range	azimuth	ll_rot_vel	ll_dv	ll_base	depth_kft
1	2016-06-08T00:01:10	KBOX	955	3	F5	88	49	17	17	10	13
2	2016-06-08T00:01:10	KBOX	956	3	F5	91	48	17	19	11	18
3	2016-06-08T00:01:14	KCXX	497	7L	D6	14	86	51	30	2	2
4	2016-06-08T00:05:22	KCXX	117	5L	D6	13	107	38	54	1	2
5	2016-06-08T00:08:07	KDEN	183	3	X1	20	242	14	14	5	19
6	2016-06-08T00:08:55	KRIW	614	7L	B6	34	276	48	41	3	6
7	2016-06-08T00:09:20	KEPZ	788	6	A6	76	33	31	62	8	11
8	2016-06-08T00:10:20	KBOX	956	3	F5	93	49	14	20	11	18
9	2016-06-08T00:10:59	KDEN	203	3	X1	20	242	17	16	5	19
10	2016-06-08T00:10:59	KDEN	204	3	X1	18	232	30	26	10	12

Obs	dpth_stmrl	max_rv_kft	max_rv_kts	tvs	motion_deg	motion_kts	msi	shape
1	100	14	22	N	278	14	1843	POINT (-69.6306900686946 42.9087204366999)
2	100	19	28	N	353	33	1721	POINT (-69.6019306665078 42.9609951364008)
3	47	2	63	N	-999	-999	6050	POINT (-72.84057763236 44.5268104383833)
4	44	2	49	N	311	5	4750	POINT (-72.8767138260852 44.4472967220186)
5	0	11	29	N	226	14	1819	POINT (-104.906667800608 39.5707779820797)
6	39	3	48	N	-999	-999	4293	POINT (-109.246654714611 43.1226528729933)
7	67	8	31	N	-999	-999	3187	POINT (-105.878317682481 32.9347703082943)
8	100	19	22	N	305	15	1480	POINT (-69.5437406633111 42.962259402214)
9	0	11	26	N	-999	-999	1964	POINT (-104.906667800608 39.5707779820797)
10	0	15	31	N	-999	-999	1726	POINT (-104.831639940611 39.5427723681603)

The SASENOAA engine creates a temporary data set named OUTTP1 that shows the recorded feature (mesocyclone detection algorithm) count for each BY group by WSR_ID. The count represents the number of mesocyclones detected by that weather station. This information can be helpful for determining which geographic area to focus on and is generated automatically by the engine when you specify STAT='COUNTGROUPBY:WSR_ID'. The SASENOAA engine does not save this data set unless the STAT= option is specified; this results in a saved statistics data set that is named by appending _S to the name specified in the OUTXML= option, as shown by the following statements:

```
libname mystats "U:\noaa940\test\";

proc contents data=mystats.c9nco_S; run;
proc print data=mystats.c9nco_S(obs=20); run;
```

Output 50.6.2 Digital Mesocyclone Detection Algorithm Statistics Data for June 8, 2016

Digital Mesocyclone Detection Algorithm Data for June 8, 2016

Obs	wsr_id	count
1	KOTX	120
2	KPDT	113
3	KBOX	90
4	KMSX	23
5	KCXX	22
6	KRTX	20
7	KLGX	18
8	KAMA	15
9	KPUX	13
10	KEPZ	12
11	KTLX	8
12	KCBW	6
13	KDDC	5
14	KGYX	4
15	KTFX	4
16	KRIW	3
17	KFDX	3
18	KMCO	3
19	KAMX	3
20	KDEN	3

For brevity, only the first 10 out of 525 observations are printed by using the OBS= option in the PROC PRINT statement for Output 50.6.1. The first 20 observations of the statistics data set c9nco_S are shown in Output 50.6.2.

In Example 50.8, another method is used to subset results by location when you use the TILE= option. In Example 50.7, the STAT= option is used to collect statistics based on a tile summary in a data set (Mytile_S).

Example 50.7: Retrieving Tornado Vortex Signature Data Statistics by Using Tile Summary Statistics

This example retrieves tornado vortex signature data statistics for the range from May 5 to May 16, 2009, but only returns the actual NX3TVS data for 11 days starting on May 5, 2006. **NOTE:** The NOAA SWDI web service allows a range longer than one year for statistics reporting, but it allows only up to a year for the range of data that you retrieve. The SASENOAA engine uses the specified start and end dates unless the range exceeds one year (of data retrieval). When the range exceeds one year, the SASENOAA engine issues an invalid range warning and defaults to a different end date. The new end date uses an end year that matches the start date's year. Sometimes this default behavior might generate an end date that precedes the start date, resulting in only one day (corresponding to the start date) of data retrieved for the OUTXML= options results file.

This example generates an 11-day default range when the end year is changed to 2006 (from 2009); the results in the Mytile data set are shown in Output 50.7.1. The Mytile_S data set shows the recorded feature

(tornado vortex signature) count for the specified tile in the tile summary specification, and it includes the centerlat, centerlon, day (date), fcount (feature count for that day), and shapefile. The count represents the number of tornado vortex signatures detected within the tile summary coordinates. This information can be helpful for determining which geographic area and dates to focus on.

```
options validvarname=any;

title 'Retrieve NOAA NX3TVS Tile Summary Statistics and Data for Date Range';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=nx3tvs
   range='20060505:20090516'
   stat='tilesum:-102.0,32.7'
   outXml=mytile
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\mytile.map"
   format=xml
   ;

data mylib.stattil;
   set noaa.mytile;
run;

proc contents data=mylib.stattil; run;
proc print data=mylib.stattil(obs=10); run;
```

Output 50.7.1 Retrieve NOAA NX3TVS Tile Summary Statistics and Data for Date Range

Retrieve NOAA NX3TVS Tile Summary Statistics and Data for Date Range

Obs	ztime	wsr_id	cell_id	cell_type	range	azimuth	max_shear	mxdv	shape
1	2006-05-05T00:05:50	KBMX	Q0	TVS	7	217	403	116	POINT (-86.8535716274277 33.0786326913943)
2	2006-05-05T00:10:02	KBMX	Q0	TVS	5	208	421	120	POINT (-86.8165772540846 33.0982820681588)
3	2006-05-05T00:12:34	KSJT	P2	TVS	49	106	17	52	POINT (-99.5771091971025 31.1421609654838)
4	2006-05-05T00:17:31	KSJT	B4	TVS	40	297	25	62	POINT (-101.188161700093 31.672392833416)
5	2006-05-05T00:29:13	KMAF	H4	TVS	53	333	34	111	POINT (-102.664426480293 32.7306917937698)
6	2006-05-05T00:31:25	KLBB	N0	TVS	51	241	24	78	POINT (-102.70047613441 33.2380072329615)
7	2006-05-05T00:33:25	KMAF	H4	TVS	52	334	46	145	POINT (-102.6393683028 32.7226656893341)
8	2006-05-05T00:37:37	KMAF	H4	TVS	50	334	34	107	POINT (-102.621904684258 32.6927081076156)
9	2006-05-05T00:41:51	KMAF	H4	TVS	51	335	29	91	POINT (-102.614794815627 32.714139844846)
10	2006-05-05T00:44:33	KLBB	N0	TVS	46	245	35	100	POINT (-102.643380529494 33.3266446067682)

NOTE: The date range that is specified in the RANGE= option is invalid for the OUTXML data because it spans more than one year, but the STAT= option can use the longer range (as specified) to report the tile summary statistics. For the file specified in the OUTXML= option, the SASENOAA engine issues a warning that the range is invalid, and it changes the end year to the same year as the start year in an attempt to keep the range under one year. In this example, for brevity, OBS=10 is specified in the PROC PRINT statement. You can use the following statements to generate the statistics results, which are shown in Output 50.7.2.
NOTE: The data that are shown in Output 50.7.2 are restricted only by the date range, not by the tile summary coordinates. To restrict the data results by the coordinates of a tile, use the TILE= option as shown in Example 50.8.

```
libname mystats "U:\noaa940\test\";

proc contents data=mystats.mytile_S; run;
proc print data=mystats.mytile_S; run;
```

Output 50.7.2 Tornado Vortex Signature Statistics Using the STAT=TILESUM Option

Retrieve NOAA NX3TVS Tile Summary Statistics and Data for Date Range

Obs	day	centerlat	centerlon	fcount	shape
1	2007-03-29	32.7	-102	2	POLYGON ((-102.05 32.65, -102.05 32.75, -101.95 32.75, -101.95 32.65, -102.05 32.65))
2	2007-09-07	32.7	-102	1	POLYGON ((-102.05 32.65, -102.05 32.75, -101.95 32.75, -101.95 32.65, -102.05 32.65))
3	2008-05-27	32.7	-102	4	POLYGON ((-102.05 32.65, -102.05 32.75, -101.95 32.75, -101.95 32.65, -102.05 32.65))
4	2008-06-20	32.7	-102	2	POLYGON ((-102.05 32.65, -102.05 32.75, -101.95 32.75, -101.95 32.65, -102.05 32.65))
5	2009-04-11	32.7	-102	1	POLYGON ((-102.05 32.65, -102.05 32.75, -101.95 32.75, -101.95 32.65, -102.05 32.65))

NOTE: You can get one day of results for the OUTXML= option by using an end date that is earlier than the start date specified in the RANGE= option. Furthermore, in this example, because both the specified start and end dates are in May, if the specified end date had been May 1, 2009, instead of May 16, 2009, then the statistics results would have been very similar, but the XML file would contain only the results for May 5, 2006. The SASENOAA engine forces the range to use the same year when the specified range exceeds one year. This can sometimes result in an invalid end date that precedes the start date, but the SASENOAA engine then discards the end date so that the range spans only one day, which is the start date.

Example 50.8: Retrieving Tornado Vortex Signature Data by Using Tile Coordinates

This example retrieves the tornado vortex signature (TVS) data for the range May 5 to May 16, 2006, but it selects only the data that fall inside the geographic area defined by the specified tile's longitude and latitude coordinates (to the nearest tenth of a degree). Output 50.8.1 shows five observations within range of the coordinates specified in the TILE= option.

```
options validvarname=any;

title 'Retrieve NOAA NX3TVS Using TILE= Option with a Date Range';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   noaaset=nx3tvs
   range='20060505:20060516'
   tile='-102.12,32.62'
   outXml=my2TL
   automap=replace
   mapref=MyMap
   xmlmap="U:\noaa940\test\my2TL.map"
   format=xml
   ;

data mylib.TVStil;
   set noaa.my2TL;
run;

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

Output 50.8.1 Using the TILE= Option to Retrieve TVS Data for a Date Range

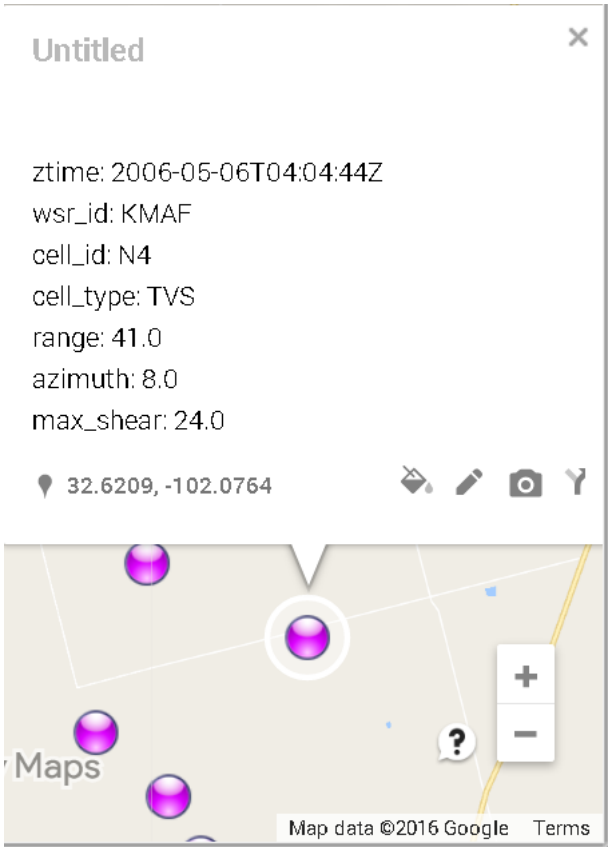
Retrieve NOAA NX3TVS Using TILE= Option with a Date Range

Obs	ztime	wsr_id	cell_id	cell_type	range	azimuth	max_shear	mxdv	shape
1	2006-05-06T00:41:29	KMAF	D9	TVS	37	6	39	85	POINT (-102.112726356403 32.5574494581267)
2	2006-05-06T03:56:18	KMAF	N4	TVS	39	3	30	73	POINT (-102.14873079873 32.5933553250156)
3	2006-05-06T03:56:18	KMAF	N4	TVS	42	4	20	52	POINT (-102.131167022161 32.6426287452898)
4	2006-05-06T04:00:30	KMAF	N4	TVS	38	5	35	86	POINT (-102.123671677514 32.5751241756203)
5	2006-05-06T04:04:44	KMAF	N4	TVS	41	8	24	62	POINT (-102.076389686189 32.6209390786829)

NOTE: You could add the option STAT='COUNTGROUPBY:WSR_ID', and the statistics would be stored in a data set named My2TL_S. The statistics results data show all the reporting weather stations by WSR_ID for the specified date range and the summary count of TVS features recorded for each station.

If you want to see a Google map of the same tile's NX3TVS data, you can rerun this example with the FORMAT=KMZ, KMZMAP=, and OUTKMZ= options to download the corresponding KML file. After you import it to Google My Maps, you see a map like the one shown in Output 50.8.2. When you click on the rightmost data point, you can examine the details of that particular location on the map.

Output 50.8.2 Screen Shot of Google Earth Map of the NX3TVS Data for TILE=-102.12,32.62



Example 50.9: Mapping Hail Data in a Geospatial Framework (KMZ Format) for a Specific Weather Station

This example retrieves the same hail data as in Example 50.4, but instead of requesting the XML format, it requests the KMZ format, so that you can look at the data in a geospatial framework such as that provided by Google Maps.

```
options validvarname=any;

title 'Retrieve NOAA NX3HAIL Data for WSR_ID=KFWS on May 21, 2011';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   debug=on
   noaaset=nx3hail
   range='20110521:20110522'
   filterBy='WSR_ID:KFWS'
   filterByCondition='WSR_ID:or'
   outkmz=myK2by
   automap=replace
```

```
   mapref=MyMap
   kmzmap="U:\noaa940\test\myK2by.kmz"
   format=kmz
   ;

data mylib.HAILby2;
   set noaa.myK2by;
run;

proc contents data=mylib.HAILby2; run;
proc print data=mylib.HAILby2; run;
```

Output 50.9.1 Using FORMAT= KMZ Option to Retrieve NX3HAIL Data for WSR_ID:KFWS

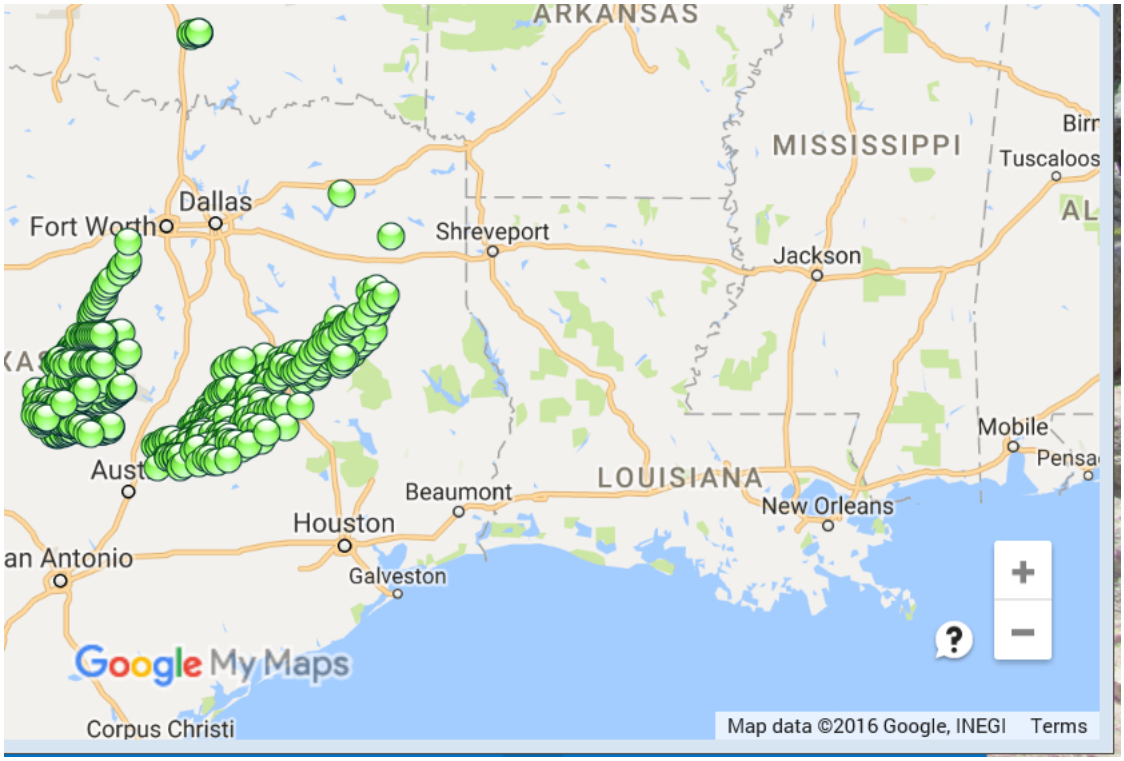
Files in the ZIP file

Obs	memname	isFolder	memcount
1	swdi-export.kml	0	1

NOTE: The KMZ file shown in Output 50.9.1 is automatically unzipped and renamed *MYK2BY.kml* by the SASENOAA engine.

Output 50.9.2 shows the Google Earth map for the observations within range of the weather station designated by the filter WSR_ID=KFWS. When you import your KML file (*MYK2BY.kml*) into Google My Maps, you can examine the details of each set of mapped coordinates on the map by clicking on the data point you want to look at.

Output 50.9.2 Screen Shot of Google Earth Map of the NX3HAIL Data in MYK2BY.kml



Example 50.10: Mapping Hail Data in a Geospatial Framework (SHP Format) for a Specific Weather Station

This example retrieves the same hail data as in Example 50.9, but instead of requesting the KMZ format, it requests the SHP format, so that you can look at the data in a geospatial framework such as that provided by Esri mapping software. Output 50.10.1 shows the retrieved Esri shapefiles that contain the data for the observations within range of the weather station designated by the filter WSR_ID=KFWS.

```
options validvarname=any;

title 'Retrieve NOAA NX3HAIL Data for WSR_ID=KFWS on May 21, 2011';
libname _all_ clear;
libname mylib "U:\noaa940\doc\";

libname noaa sasenoaa "U:\noaa940\test\"
   debug=on
   noaaset=nx3hail
   range='20110521:20110522'
   filterBy='WSR_ID:KFWS'
   filterByCondition='WSR_ID:or'
   outshp=mySby
   automap=replace
   mapref=MyMap
   shpmap="U:\noaa940\test\mySby.map"
   format=shp
   ;

data mylib.HAILbyS;
   set noaa.mySby;
run;

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

Output 50.10.1 Using FORMAT= SHP Option to Retrieve NX3HAIL Data for WSR_ID:KFWS

Files in the ZIP file

Obs	memname	isFolder	memcount
1	swdi-nx3hail-all-20161021-110558-991.dbf	0	4
2	swdi-nx3hail-all-20161021-110558-991.prj	0	4
3	swdi-nx3hail-all-20161021-110558-991.shp	0	4
4	swdi-nx3hail-all-20161021-110558-991.shx	0	4

NOTE: The SASENOAA engine automatically unzips the ZIP file that contains the four shapefiles and renames them *MYSBY.dbf*, *MYSBY.prj*, *MYSBY.shp*, and *MYSBY.shx*.

References

Johnson, J. T., MacKeen, P. L., Witt, A., Mitchell, E. D., Stumpf, G. J., Eilts, M. D., and Thomas, K. W. (1998). "The Storm Cell Identification and Tracking Algorithm: An Enhanced WSR-88D Algorithm." *Weather and Forecasting* 13:263–276.

National Centers for Environmental Information (2016). "NCEI Data Access." Accessed August 4, 2016. http://www.ncdc.noaa.gov/data-access.

National Oceanic and Atmospheric Administration (2016a). "Multi-radar Multi-sensor System Web Application Suite Launcher." Accessed August 5, 2016. http://mrms.ou.edu/.

National Oceanic and Atmospheric Administration (2016b). "NOAA National Centers for Environmental Information (formerly National Climatic Data Center)." Accessed August 4, 2016. http://www.ncdc.noaa.gov/?datasetname=7000.

National Oceanic and Atmospheric Administration (2016c). "Severe Weather Data Inventory (SWDI) REST Web Services Usage." Accessed August 4, 2016. https://www.ncdc.noaa.gov/swdiws/.

Subject Index

Syntax Index