PROC GEOCODE: Finding Locations Outside the U.S.
Darrell Massengill and Ed Odom, SAS Institute Inc., Cary, NC

ABSTRACT
How do you convert addresses into map locations? This is done through the process of geocoding. PROC GEOCODE was first included in SAS/GRAph 9.2 to provide this capability. Street-level geocoding for the United States was later added to the third maintenance release of SAS 9.2 (9.2M3). This paper will review all of the capabilities of this procedure including the new abilities to geocode international cities added in SAS 9.3M2 and Canadian street-level geocoding available in SAS 9.4. Also, you can now import free postal code data for Great Britain and Australia for geocoding in all releases of PROC GEOCODE.

CONTENTS
Abstract
Introduction
Geocoding Basics
City Location
Postal Code Location
ZIP+4 Location
Street-level Location
IP Address Location
Geocoding Method Selection
PROC GEOCODE Overview
Procedure Syntax
Lookup Data
  CITY
    SAS 9.2 - 9.3M1
    SAS 9.3M2 - 9.4
  ZIP
    U.S. ZIP codes
    Great Britain Postcodes
    Australian Postcodes
    Canadian Postcodes
    Other Countries
  ZIP+4
    From SAS
    From Melissa Data
  STREET
    SAS 9.2 - 9.3M2
    SAS 9.4
  RANGE
  CUSTOM
Examples
  CITY
    SAS 9.2 - 9.3M1
    SAS 9.3M2 - 9.4
  ZIP
    U.S. ZIP Codes
    Great Britain Postcodes
    Australian Postcodes
  ZIP+4
    From SAS
    From Melissa Data
  STREET
    United States
    Canada
  IP Address
  CUSTOM
Summary
Resources
Contact Information
Appendix 1 - ZIP Code FAQ
Appendix 2 - SAS Spatial Capability Summary
INTRODUCTION

Every organization has large amounts of data that include location components ranging from a city name, postal code or a complete mailing address. The address data is useful only if it is transformed into a geographic location that can be viewed on a map, used in distance calculations or various spatial analyses. To make this data useful, you must convert the address to a location by determining its latitude and longitude. This conversion process is called geocoding. For IP addresses the process is also known as geolocation.

This paper will discuss geocoding using the SAS/GRAPH GEOCODE Procedure. First, we will introduce the concepts needed to understand geocoding, and then we will discuss PROC GEOCODE’s traditional and new functionality. Finally, examples will show how to use PROC GEOCODE.

GEOCODING BASICS

The geocoding process depends on lookup data with the necessary information to convert an address to a geographic location. This data is the key to geocoding. Factors such as age and granularity of the lookup data determine the geocoding results. Addresses routinely change because of commercial or residential construction, new streets, and postal codes being split and changed. As a general rule, use the most recent lookup data containing new streets and the latest corrections. However, when geocoding data from a specific year, it may be better to use lookup data for that year.

Granularity is another important consideration. Does the location need to be the actual house position or will a more generalized location be close enough? For example, when viewing geocoded addresses on a state or U.S. map, a ZIP code or city centroid is accurate enough. There is no visible difference between a point placed on Elm Street and another on Turner Drive in Houston when viewed on a statewide map of Texas.

To understand geocoding, it is important to first understand the lookup data. It is particularly important to understand the differences between ZIP code data, ZIP+4 data, street address data and city center data. IP address data is completely different from the other types of addresses, but it is important to understand this data, too.

Another factor to consider when choosing a geocoding method is your input data. What address components does it contain? For example, do you have the complete mailing address including house number, street name, city, state and ZIP code, or do you have only parts of the full address? Also, the quality and consistency of your data impacts the geocoding process. If your addresses contain extraneous information such as suite or floor numbers, it may cause parsing difficulties for the geocoder.

CITY LOCATION

Of the various geocoding methods, the most generalized location is by city name. This is perfectly adequate for locating points over a large geographic area, for example addresses spanning a large nation such as Canada.

City geocoding requires more than just a city name. At a minimum the city and state or province names are needed as city names are not unique. For example there are three cities named San Francisco within the U.S. Without specifying a state, it is impossible to determine which one is wanted. For that same reason, a country name is also required when geocoding international cities as there are at least 88 cities named San Francisco across the globe.
POSTAL CODE LOCATION

After city locations, the next more precise geocoding method is by postal codes which are used in many countries to define mail delivery routes or locations. The codes vary in length and can contain a mix of alphabetic characters, numeric digits and spaces. In the United States, postal codes are called ZIP codes and contain five digits. In this paper, ZIP code and postal code are used interchangeably.

While most ZIP codes apply to a set of specific streets in an area, some are assigned to a single building or to a post office. Boundaries between ZIP codes are not created by the U.S. Postal Service (USPS). ZIP codes are not officially defined polygonal areas in the manner of a county or city boundary. Creating polygons by simply enclosing the linear delivery routes would leave gaps between the polygons because there are large, undeveloped areas.

The area covered in a ZIP code varies by its population density. A rural ZIP code covers a larger area than one in a city. As noted, some ZIP codes apply to only a single building. Generally, ZIP code address data specifies a centroid location for the ZIP code area. Because those ZIP boundaries are generated by different data vendors, their centroid locations often vary. Appendix 1 discusses frequently asked questions about ZIP codes.

Figure 1 illustrates an artificial boundary created to enclose the streets in a ZIP code. When geocoding addresses having this ZIP code, all are assigned the latitude and longitude of the polygon’s centroid. This behavior also applies to non-U.S. postal codes.

Figure 1. Boundary Around Streets with Common ZIP Code

So, when geocoding an address by its ZIP code, the location will be in the general vicinity of the address, but likely not on the actual street.
ZIP+4 LOCATION

A U.S. ZIP code is also subdivided into smaller delivery routes which are tagged with a ZIP+4 identifier. A hyphen and four additional digits are appended to the ZIP code to specify these additional subdivisions. A ZIP+4 will likely represent a single street or a part of a street. In a high-density urban area, it might represent one side of a street on a single block or even one floor of a large building. Figure 2 illustrates the relationship between a ZIP code and one of its ZIP+4 segments. The ZIP+4 centroid is at the midpoint of that street segment.

Centroids are computed for each ZIP+4 in a ZIP code. Any address within a ZIP+4 would be assigned that centroid if geocoded by ZIP+4 data.

While a ZIP code will get you to the general vicinity of an address, ZIP+4 geocoding will probably get you to the correct street in the address.
STREET-LEVEL LOCATION

The most precise location method is street-level geocoding using the full mailing address: house number, street name, and either ZIP code or city and state. This does not necessarily place the geocoded address at the exact house location. It approximates the position of a particular address on a street by assuming that house numbers are an equal distance apart, which might not be true. But it is still a more precise location than placing the geocoded point at the centroid of a ZIP code or city.

As Figure 3 illustrates, a street-level location is placed along the given street as near the exact structure as possible given the limitations of the lookup data. If you have ever looked up your home address on the Web, it is likely that the location did not align exactly with your house. That precision is a characteristic of the lookup data available. However, the geocoded location was close enough for most uses.

![Figure 3. Location using House Number and Street Name](image)

IP ADDRESS LOCATION

Unlike street addresses, IP addresses were not designed to be geographic. Generally, these are collected from visitors to Web sites and indicate the connection the visitor used. IP address lookup data contains information that matches ranges of IP addresses to particular geographic locations. The location found will not be at the street or even ZIP code level, but might indicate the city, state, or country where the IP address is registered.
GEOCODING METHOD SELECTION

Several factors must be considered when choosing the appropriate type of geocoding:

1. Geographic Extent
   If the data to be geocoded spans a large region, the extra precision of street-level geocoding may not be required. ZIP code or possibly even city geocoding may be sufficient for your needs.

2. Attribute Values Wanted
   Do you want to assign values to your data based on the geocoded locations? For example, the street geocoding method can also assign Census Tract numbers to your addresses.

3. Location Precision
   This is related to both the Geographic Extent and Attribute Values Wanted parameters discussed above. More precise locations may be required when your data is in a smaller area or if you want to assign attribute values that require precise locations.

4. Address Components Present
   The geocoding method also depends on the content of your input data. If your data lacks street names, then street-level geocoding is not possible. You would have to augment your input data or geocode it by ZIP or city.

5. Lookup Data Availability
   You must have the appropriate lookup data covering the geographic extent of your addresses. For example, if you do not have postal code or street-level lookup data for a region, you cannot use those geocoding methods for addresses there. The cost and disk space required for the lookup data can also be a factor.

PROC GEOCODE can also do multiple types of geocoding in one run. For example, if the STREET method is specified as the primary geocoding process and an address is not matched, the geocoder will pass that address to the ZIP method. If the ZIP method cannot match the address, the CITY method is then tried. This cascading into the next geocoding level is the default behavior which can be disabled if desired.
PROC GEOCODE Overview

The GEOCODE procedure computes geographic coordinates (latitude and longitude values) from address data. These geographic coordinates can then be plotted on a map, used for distance calculations or in spatial analyses. Appendix 2 contains more information about what can be done with the geocoded coordinates. In addition, the procedure enables you to add attribute values from your lookup data to the geocoded locations. Examples would be adding Census Tracts or area codes to a geocoded address.

The six methods of geocoding and currently available lookup data sources are listed in Table 1. Some of the lookup data are install with SAS, some can be downloaded from the SAS MapsOnline web site, some is freely available from third party providers, and some must be purchased. For the third party suppliers listed, SAS provides a program to import their data into geocoding lookup data sets.

Not all of the lookup data in Table 1 is available for all releases of PROC GEOCODE. For example, the CITY method lookup data in the MAPSGFK library was added in the second maintenance release of SAS 9.3 (9.3M2). And the Canadian STREET lookup data from GeoBase is available only for SAS 9.4 and will not work with earlier SAS releases.

<table>
<thead>
<tr>
<th>Geocoding Method</th>
<th>Input Data</th>
<th>Coverage</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>CITY</td>
<td>City and state name</td>
<td>United States</td>
<td>MAPSGFK.USCITY_ALL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>World</td>
<td>SASHELP.ZIPCODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MAPSGFK.WORLD_CITIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WORLD_CITIES_ALL</td>
</tr>
<tr>
<td>ZIP</td>
<td>Postal code</td>
<td>United States</td>
<td>SASHELP.ZIPCODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Australia</td>
<td>Australian Bureau of Statistics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>England and Scotland</td>
<td>Ordnance Survey of Great Britain</td>
</tr>
<tr>
<td>PLUS4</td>
<td>ZIP code with ZIP+4</td>
<td>United States</td>
<td>MapsOnline</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Melissa Data</td>
</tr>
<tr>
<td>STREET</td>
<td>Complete mailing address</td>
<td>United States</td>
<td>MapsOnline</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Census TIGER shapefiles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Canada</td>
<td>GeoBase National Road Network</td>
</tr>
<tr>
<td>RANGE</td>
<td>IP address</td>
<td>World</td>
<td>MaxMind</td>
</tr>
<tr>
<td>CUSTOM</td>
<td>User-defined region</td>
<td>n/a</td>
<td>User created</td>
</tr>
</tbody>
</table>

Table 1. Geocoding Methods and Lookup Data

By default if the primary method specified cannot geocode an address, the procedure attempts to geocode it with the next (less precise) method. For example, if you specify street geocoding and no match is found, the procedure will try the ZIP method. If that fails to find a match, it then tries the city method. This cascading method behavior can be turned off if desired.
The _MATCHED_ variable in the output data set indicates the type of match that was found for each address. Values are listed in the PROC GEOCODE documentation.

The GEOCODE procedure requires two types of SAS data sets:

1. The first is an input data set of addresses to geocode. This data set will contain variables related to the address such as street address, city, state, and ZIP code. Note that it is important for this data to be clean and in the proper form in order to have the best match rate.

2. The second type is one or more lookup data sets that are needed to transform your address data into geographic locations. The number of data sets depends on the geocoding method. ZIP, PLUS4 and CUSTOM geocoding each use one lookup data set, IP address geolocation uses two, and street-level references six.

The simplest example illustrating how these data sets are used is with ZIP code geocoding and its single lookup data set. Figure 4 shows that all of the variables from your input data are carried forward into the output data set. The ZIP variable as the geocoding key is found up in the lookup data set and those LONG and LAT values are added to the output data set. These are the geographic coordinates of that ZIP code centroid. In addition, if the lookup data set contains other attributes such as county names or Census Blocks, you can specify that these additional values also be moved to the output data set. The processing for other geocoding methods is a bit more complicated, especially street level.

![Figure 4. ZIP Code Lookup Process](image-url)
PROCEDURE SYNTAX

The syntax for the GEOCODE procedure is very flexible and enables you to specify different lookup data sets and different variable names in case the defaults do not match your data names.

The basic syntax is like other SAS procedures:

PROC GEOCODE <options>;

Options applying to various data sets include:

DATA= points to the address data set containing the addresses to be geocoded. The options in Tables 2 and 3 that begin with ADDRESS are used to specify the names of the variables in this data set. If the variable names match the DEFAULT column in the table, the option is not required.

The OUT= option specifies an output data set for the geocoded results. If not specified, SAS will generate a data set name automatically.

LOOKUP applies to several option variants which specify lookup data sets for different geocoding methods. Other options in the table beginning with LOOKUP are used to specify the names of the variables in this data set. If the variable names match the DEFAULT column, then the option is not required. The RANGE and STREET geocoding methods require multiple lookup data sets. This is explained in more detail later.

The METHOD= option selects the primary type of geocoding. The method specified controls the type of geocoding performed and the lookup data required. Supported methods are:

- CITY – City and state geocoding. The default lookup data set depends on whether your cities are all within the U.S. or are international.
- ZIP – ZIP code or non-U.S. postal code centroid geocoding. If no match is found for a ZIP code, the geocoder then uses the CITY method unless the NOCITY option is specified. The default lookup data set is SASHELP.ZIPCODE.
- PLUS4 – ZIP+4 geocoding. If no match is found for the nine-digit ZIP+4, then it uses the ZIP method unless the NOZIP option is specified.
- STREET – This method converts a full street address that includes a house number, street name, city, state, and ZIP code to a geographic location. Every effort will be made to match the full address as closely as possible, but if it fails then the ZIP method is performed. As with straight ZIP geocoding, if no ZIP match is made, the geocoder then attempts a city match. NOZIP and NOCITY disable this cascading behavior. This method requires five or six lookup data sets and additional options which are explained later.
- RANGE – Matches are made to lookup data containing a range of values. IP address geolocation is one example. This method requires specifying a second data set with the RANGEDATA= option in addition to the LOOKUP= option.
- CUSTOM – This method enables you to use custom lookup data that you generate. Examples are internal sales territories, Metropolitan Statistical Areas (MSA), and telephone area codes.

Different options are used with the various geocoding methods. The following tables indicate which options apply to each geocoding method. Table 1 lists the options available through the 9.3M1 release. Table 2 notes options added in SAS 9.3M2. The DEFAULT column indicates the variable or data set name used if the option is omitted from the PROC GEOCODE syntax.

The complete syntax and detailed descriptions of the options are in the SAS/GRAPH Reference for the GEOCODE Procedure. The GEOCODE reference for your specific SAS release will list all options available at your site.
<table>
<thead>
<tr>
<th>Option</th>
<th>CITY</th>
<th>ZIP</th>
<th>PLUS4</th>
<th>STREET</th>
<th>RANGE</th>
<th>CUSTOM</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESSCITYVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>CITY</td>
</tr>
<tr>
<td>ADDRESSPLUS4VAR=</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>PLUS4</td>
</tr>
<tr>
<td>ADDRESSSTATEVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>STATE</td>
</tr>
<tr>
<td>ADDRESSVAR=</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ADDRESSZIPVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>ZIP</td>
</tr>
<tr>
<td>ATTRIBUTEVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>BEGINRANGEVAR=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Previous Data Set Used</td>
</tr>
<tr>
<td>ENDRANGEVAR=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIPS=</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SASHELP.PLFIPS</td>
</tr>
<tr>
<td>LOOKUP=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>LOOKUPCITYVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>CITY</td>
</tr>
<tr>
<td>LOOKUPKEYVAR=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOOKUPLUS4VAR=</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>PLUS4</td>
</tr>
<tr>
<td>LOOKUPSTATEVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>STATECODE</td>
</tr>
<tr>
<td>LOOKUPSTREET=</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOOKUPVAR=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOOKUPXVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>LOOKUPYVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Y</td>
</tr>
<tr>
<td>LOOKUPZIPVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>ZIP</td>
</tr>
<tr>
<td>METHOD=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>ZIP</td>
</tr>
<tr>
<td>NOCITY</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOSTIMER</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NOZIP</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUT=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>DATA*n</td>
</tr>
<tr>
<td>RANGEDATA=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RANGEDECIMAL</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RANGEKEYVAR=</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. PROC GEOCODE Options Through SAS 9.3M1
### Table 3. PROC GEOCODE Options Added in SAS 9.3M2

<table>
<thead>
<tr>
<th>Option</th>
<th>CITY</th>
<th>ZIP</th>
<th>PLUS4</th>
<th>STREET</th>
<th>RANGE</th>
<th>CUSTOM</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESSCOUNTRYVAR=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOOKUPCITY=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MAPSGFK.USCITY_ALL MAPSGFK.WORLD_CITIES</td>
</tr>
<tr>
<td>LOOKUPCOUNTRYVAR=</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ISOALPHA2 ISOALPHA3</td>
</tr>
<tr>
<td>LOOKUPLATVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>LAT</td>
</tr>
<tr>
<td>LOOKUPLONGVAR=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>LONG</td>
</tr>
</tbody>
</table>

### LOOKUP DATA

Geocoding lookup data is installed for several of the methods. However, some lookup data sets are too large to ship with SAS or must be acquired directly from third parties. Macro code programs are provided to import third-party data. All of the lookup data currently available is described below for each geocoding method and applicable SAS version.

It is also important to remember that both purchased and free lookup data contain data entry errors which can cause missed matches or incorrect results. There are no guarantees with any geocoding lookup data, so your results should be reviewed carefully.

- **CITY**

  Geographic coverage for the CITY geocoding method depends on your SAS release.

  **SAS 9.2 – 9.3M1**
  - Geocodes only cities within the U.S.
  - Lookup data set is SASHELP.ZIPCODE
  - Installed with SAS and updated each release
  - Quarterly updates also available from [MapsOnline](#)

  **SAS 9.3M2 – 9.4**
  - Geocodes U.S. and international cities
  - Default U.S. lookup data set is MAPSGFK.USCITY_ALL
  - Default international lookup data set is MAPSGFK.WORLD_CITIES
  - Both default data sets are installed with GRAPH in the MAPSGFK library and updated each release
  - SASHELP.ZIPCODE can also still be used as an alternate lookup data set for U.S. cities
  - Optional WORLD_CITIES_ALL lookup data set also available from MapsOnline
  - It contains many more world cities than the default MAPSGFK.WORLD_CITIES data set
• ZIP

Postcode lookup data is available by country from SAS and third party data vendors.

U.S. ZIP codes

- Default lookup data set is SASHELP.ZIPCODE
- Installed with SAS and updated each release
- Quarterly updates also available from MapsOnline

Great Britain Postcodes

- Free Royal Mail postcode location files from British Ordnance Survey
- Coverage includes England and Scotland
- Use %CODEPOINT2GEOCODE macro program to import CSV file
- Import program and documentation available from MapsOnline

Australian Postcodes

- Free Australia Post postcode polygons from Australian Bureau of Statistics
- Available as Postal Area (POA) file
- POA file contains postcode polygons for Australia
- %ABS2GEOCODE macro program imports POA shapefile and computes centroids
- Import program and documentation available from MapsOnline

Canadian Postcodes

- Purchase Premium Edition product from data vendor ZIP Code Download
- CSV file contains Canadian postcode centroids
- MapsOnline provides import macro %ZIPCODEDOWNLOAD2GEOCODE to create lookup data

Other Countries

- Various spatial data vendors can provide postal code data for selected countries
- One example is MapMechanics
- Locate additional data suppliers with an Internet search
- Use PROC IMPORT, PROC MAPIMPORT or the DATA step to import centroid files
- Contact SAS Technical Support for assistance with a specific format

• ZIP+4

ZIP+4 lookup data applies only to the United States and is available from SAS and third party data vendors.

From SAS

- Free ZIP+4 lookup data set from MapsOnline
- Created from TIGER 2006, Second Edition data
- TIGER releases through 2012 do not include ZIP+4 values
- MapsOnline lookup data set will be updated when Census Bureau restores ZIP+4 values to TIGER data

From Melissa Data

- Geo*Data product contains ZIP+4 centroids
- Import Geo*Data files with SAS autocall macro %GCDMEL9
- See PROC GEOCODE section of SAS/GRAPH documentation for macro usage
• STREET

Geographic coverage and lookup data format vary by SAS release. When downloading street lookup data or an import program from MapsOnline, make sure to get the version appropriate for your SAS release.

SAS 9.2 - 9.3M2

- Geographic coverage includes U.S. states plus Puerto Rico and Virgin Islands
- Nationwide U.S. lookup data sets created from Census Bureau TIGER/Line shapefiles are available from MapsOnline
- Individual U.S. county TIGER files can also be downloaded from the Census Bureau
- The macro program %TIGER2GEOCODE available from MapsOnline will import the downloaded county TIGER files to generate street lookup data sets for those areas

SAS 9.4

Geographic coverage includes U.S. states, Puerto Rico, the Virgin Islands and Canadian provinces.

- United States
  - Nationwide U.S. lookup data sets from Census Bureau TIGER/Line shapefiles available from MapsOnline
  - Individual U.S. county TIGER files can also be downloaded from the Census Bureau
  - The macro program %TIGER2GEOCODE available from the above MapsOnline link will import the county TIGER files to generate street lookup data sets

- Canada
  - Canadian lookup data sets created from GeoBase National Road Network (NRN) shapefiles are available from MapsOnline
  - Individual Canadian province NRN files can also be downloaded for free directly from GeoBase
  - The macro program %GEOBASE2GEOCODE available from the above MapsOnline link will import the province NRN files to generate street lookup data sets for the downloaded provinces

Also, sample lookup data sets for Wake County, N.C. are installed in the SASHELP library: GEOEXM, GEOEXS and GEOEXP. These allow you to try out PROC GEOCODE before installing the large lookup data sets noted above. The STREET method example in this paper uses these lookup data sets.

• RANGE

The RANGE method supports geolocation of IP addresses at the country or city level. Currently only IPv4 addresses are supported.

- Lookup data available from MaxMind
- Free and paid versions of each level
- Paid versions “are more accurate”
- Download free data in CSV format:
  - GeoLite Country
  - GeoLite City
- Import with %MAXMIND autocall macro
- See PROC GEOCODE section of SAS/GRAPH documentation for macro usage

• CUSTOM

Any geographic point data can be used for custom lookup data. The only requirement is that you have variables that represent longitude and latitude values plus a third variable used as the look up key. The CUSTOM geocoding method example in this paper shows how a custom lookup data set can be created.
EXAMPLES

These examples demonstrate each of the current geocoding methods available in SAS 9.4. Note that some of the examples will not run in earlier releases as PROC GEOCODE is continually being enhanced with new capabilities. It is noted if an example is version-specific.

The complete example programs are available for download from the SAS Global Forum site. Search for the title of this paper on that site. In some cases, lookup data must be downloaded and imported before running an example program.

Many of the examples plot the geocoded points onto a map. Some of the plotting programs are in the file GMAP_MACROS.sas included in the zipped file of geocoding examples. Other map plots used techniques covered in the various SAS/GRAPH papers listed in the RESOURCES section of this paper.

Table 3 is an example of U.S. address data for geocoding examples with the CITY, ZIP and PLUS4 methods.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Address</th>
<th>ZIP</th>
<th>ZIP+4</th>
<th>City</th>
<th>State</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. Cheever Loophole</td>
<td>136 E. Water St</td>
<td>19901</td>
<td>3630</td>
<td>Dover</td>
<td>DE</td>
<td>$985.33</td>
</tr>
<tr>
<td>Cuthbert J. Twillie</td>
<td>760 Moose Lodge Road</td>
<td>19934</td>
<td>2220</td>
<td>Camden</td>
<td>DE</td>
<td>$2,533.25</td>
</tr>
<tr>
<td>Kaspar Gutman</td>
<td>4701 Limestone Road</td>
<td>19808</td>
<td>1927</td>
<td>Wilmington</td>
<td></td>
<td>$33.44</td>
</tr>
</tbody>
</table>

Table 4. Sample Input Data Set

- CITY

The geographic coverage for CITY geocoding depends on your SAS release. Initial versions of PROC GEOCODE located only cities within the United States. International city geocoding was later added.

All of the following examples of CITY geocoding use the default lookup data set appropriate for the address data. Although the default lookup data set does not have to be included in the PROC GEOCODE syntax for these examples, it is shown in the syntax for clarity.

SAS 9.2 - 9.3M1

These releases support CITY geocoding within the U.S. only. This example uses the default lookup data set SASHELP.ZIPCODE to compute X/Y of each city center. In addition, the attribute values MSA and AREACODE from the lookup data set are added to the output data for each geocoded observation.

The procedure syntax is:

```sas
proc geocode
    method = city /* City method */
    data = work.customers /* Address data to geocode */
    out = work.geocoded /* Geocoded output data set */
    lookup = sashelp.zipcode /* Default lookup data set */
    attribute_var = (msa areacode); /* Attribute vars to add */
run;
```

The complete code for this program is Example 1 in the sample file named CITY_GEOCODE_US.sas. In addition to geocoding the customer locations, the sample also includes a call to the SAS macro program %DELWARE_MAPS which plots the points on a map of Delaware using the GMAP procedure as shown in Figure 5.
The second maintenance release of 9.3 SAS (9.3M2) included new lookup data sets for U.S. and international CITY geocoding. This example uses the MAPSGFK.USCITY_ALL lookup data set to geocode the U.S. cities in the sample input data set in Table 3. In addition, the attribute values from the COUNTY_NAME variable are added to the output data for each found city.

The procedure syntax for U.S. CITY geocoding is:

```sas
proc geocode
  method = city               /* City method              */
  data   = work.customers     /* Address data to geocode */
  out    = work.geocoded      /* Geocoded output data set */
  lookupcity    = mapsgfk.uscity_all /* Default lookup data set */
  attribute_var = (county_name);  /* Attribute var to add       */
run;
```

The complete code for this program is Example 2 in the CITY_GEOCODE_US.sas file. The sample also calls the `%DELAWARE_MAPSGFK` macro to plot the points on a Delaware map as in Figure 5.

![Figure 5. Geocoded Locations in Delaware](image)

This next example uses the worldwide data set MAPSGFK.WORLD_CITIES added in SAS 9.3M2 for international CITY geocoding. The example computes LONG/LAT for SAS offices around the world. Additionally, the attribute values from the CTTYPE variable are added to the data for each found city.

When geocoding non-U.S. cities, the ADDRESCITYVAR= and ADDRESSCOUNTRYVAR= options are required. These identify the variables in your address data set which contain the city and country values for each address observation. They also tell the geocoder that you are processing non-U.S. cities.

The ADDRESSSTATEVAR= option is not required. However, it is best to include it if possible. It is used when multiple matches for the city and country variables are found in the lookup data. For example, there are two Canadian cities named ‘Vanier’ which are in separate provinces, one in Quebec and the other Ontario. By
including ADDRESSSTATEVAR=STATE in the syntax, the geocoder determines which city is wanted by comparing the MapIDName (state/province/region) value in the lookup data set with the input observation.

The procedure syntax for international CITY geocoding is:

```sas
proc geocode
  method = city /* City method */
  data = SASoffices /* Addresses to geocode */
  out = work.geocoded /* Geocoded output data set */
  lookupcity = mapsgfk.world_cities /* Default lookup data set */
  addresscityvar = city /* Required city name var */
  addresscountryvar = countryID /* Required country name var */
  addressstatevar = state /* Optional state name var */
  attributevar = (cttype); /* Attribute var to add */
run;
```

The complete code for this program is in the `CITY_GEOCODE_World.sas` file. The sample also calls the `%WORLD_MAPSGFK` macro to plot the points on a world map as in Figure 6.

Not all SAS offices in the example will be located. Some are not found because the geocoder cannot resolve multiple entries found in the lookup data set. For example, the SAS office in Alexandria, Virginia, U.S. is not found because the GfK lookup data has multiple observations in Virginia by that name. Most of the non-U.S. mismatches occur because the lookup data contains a non-U.S. spelling of the city name. For example Taipai, Taiwan and Moscow, Russia are not located because the GfK lookup data uses the local spellings of T'ai-peh-shih and Muskva.

![SAS Offices](image)

**Figure 6. SAS Offices Geocoded by City and Country**
• ZIP

The geographic coverage for the ZIP geocoding method depends on your lookup data. SAS provides lookup data for U.S. ZIP codes. We also provide macro programs to import free lookup data for Royal Mail postcodes in England and Scotland as well as postcodes used by the Australia Post.

U.S. ZIP Codes

This example uses the ZIP geocoding method with the SASHELP.ZIPCODE lookup data set to geocode the sample address data from Table 3. The geocoder attempts to find the location of the ZIP code centroid for each address. If the ZIP code is not found, then PROC GEOCODE cascades into the CITY method by default. In addition, the attribute variables MSA and AREACODE will be added to the output data for each ZIP match.

The procedure syntax is:

```sas
proc geocode
  method = zip /* ZIP method */
  data = work.customers /* Address data to geocode */
  out = work.geocoded /* Geocoded output data set */
  lookup = sashelp.zipcode /* Default lookup data set */
  attribute_var = (msa areacode); /* Attribute vars to add */
run;
```

The complete code for this program is in the sample file named ZIP_GEOCODE_US.sas. In addition to geocoding the customer locations, the sample also includes a call to the SAS macro program %DELWARE_MAPS which plots the points on a map of Delaware using the GMAP procedure as shown in Figure 5.

Great Britain Postcodes

This example uses the ZIP method to geocode automobile dealerships in Great Britain using Royal Mail postcodes. The lookup data of postcode centroids is created from the Code-Point Open product available for free from the Ordnance Survey (OS) of Great Britain.

The SAS macro program %CODEPOINT2GEOCODE imports Code-Point Open CSV files and converts the centroids’ British National Grid coordinates to World Geographic System (WGS84) latitude and longitude. This program and a detailed instruction page are in the CODEPOINT2GEOCODE.zip file on SAS MapsOnline.

Example syntax to invoke the compiled Code-Point import macro follows. You would obviously need to edit the path names in the macro parameters to match locations on your system:

```sas
%CodePoint2Geocode(
  pathin = C:\Code-Point Open\Data, /* Location of Code-Point CSV files */
  pathout = C:\Code-Point Open\Lookup, /* Location to write output data set */
  out = lookup.postcodes) /* Name of lookup data set created */
```
Please note that the syntax for British postal code geocoding differs somewhat depending on your SAS release.

**SAS 9.2 - 9.3M1**

The following syntax includes the NOCITY option. If the ZIP method fails to match an observation, PROC GEOCODE attempts to use the CITY method by default. However, the world city lookup table did not ship until the second maintenance release of SAS 9.3 (SAS 9.3M2). Therefore, use the NOCITY option to disable this method.

```sas
proc geocode
    method = zip                /* ZIP method for postcodes */
    data = dealers            /* Address data to geocode */
    out = geocoded           /* Geocoded output data set */
    lookup = lookup.postcodes /* GB postcode lookup data */
    addresszipvar = postcode  /* Address data postcode var */
    lookupzipvar = pc          /* Lookup data postcode var */
    nocity;                     /* Disable CITY method */
run;
```

If your British lookup data was created using the original version of the `%CODEPOINT2GEOCODE` macro, the latitude and longitude variables for the WGS84 datum are named LAT_WGS84_DD and LON_WGS84_DD. In that case you will have to add the following options to the above syntax:

```sas
lookupxvar = Lon_WGS84_DD       /* Longitude in lookup data */
lookupyvar = Lat_WGS84_DD       /* Latitude in lookup data */
```

If you used the current version of `%CODEPOINT2GEOCODE` to import the British postal code files, the location variables use the default names of X and Y and it is not necessary to specify them in the PROC GEOCODE syntax.

The input data preparation and geocoding program are in the sample file named ZIP_GEOCODE_GB_930.sas. Although not included in this example, the geocoded locations were also plotted on a map of Great Britain with the GMAP procedure as shown in Figure 7.

**SAS 9.3M2 – 9.4**

As noted, the second maintenance release of 9.3 SAS (9.3M2) added the MAPSGFK.WORLD_CITIES data set to support international CITY geocoding. It allows PROC GEOCODE to cascade into the CITY method for any observations not located by postal code. Therefore the following syntax includes options needed if CITY geocoding is invoked:

```sas
proc geocode
    method = zip                /* ZIP method for postcodes */
    data = dealers            /* Address data to geocode */
    out = geocoded           /* Geocoded output data set */
    lookup = lookup.postcodes /* GB postcode lookup data */
    addresszipvar = postcode  /* Postcode var for ZIP method */
    addresscityvar = city     /* City var for CITY method */
    addressstatevar = borough /* County var for CITY method */
    addresscountryvar = country/* Country var for CITY method */
    lookupzipvar = pc          /* Lookup data postcode var */
run;
```

ADDRESSCITYVAR, ADDRESSSTATEVAR and ADDRESSCOUNTRYVAR are specified so any dealerships not located by postal code will be located by the CITY method. Since the addresses are not in the U.S., those variables are used if an observation cascades from ZIP to CITY geocoding. MAPSGFK.WORLD_CITIES is the default international city lookup data set. See the GEOCODE documentation for details on non-U.S. CITY geocoding. If you wish to disable this behavior you can omit those options and add NOCITY to your PROC GEOCODE syntax.
The input data preparation and geocoding program are in the sample file named ZIP_GEOCODE_GB_940.sas. Although not included in this example, the geocoded locations were also plotted on a map of Great Britain with the GMAP procedure as shown in Figure 7.
Australian Postcodes

This example geocodes rugby stadium addresses using Australian four-digit postcodes with the ZIP method. The lookup data of postcode centroids is generated using free Postal Area (POA) files from the Australian Bureau of Statistics (ABS).

The %ABS2GEOCODE macro program imports the POA files, creates a map data set of the postal areas and computes the polygon centroids. The macro program and instructions are in the ABS2GEOCODE.zip file on SAS MapsOnline.

Sample syntax to invoke the compiled POA import macro follows. The example path names would have to be edited to match locations on your system:

```sas
%ABS2Geocode(
  file = C:\POA_2011_AUST.shp, /* Downloaded ABS POA shapefile */
  pathout = C:\Lookup, /* Output location of lookup data */
  out = lookup.postcodes) /* Lookup data set created */
```

Please note that the PROC GEOCODE syntax for Australian postal code geocoding differs somewhat depending on your SAS release.

SAS 9.2 - 9.3M1

The syntax for these SAS releases includes the NOCITY option. If the ZIP method fails to match an observation, PROC GEOCODE attempts to use the CITY method by default. However, the world city lookup table did not ship until the second maintenance release of SAS 9.3 (SAS 9.3M2). Therefore, use the NOCITY option to disable the CITY method:

```sas
proc geocode
  method = zip /* ZIP method for postcodes */
  data = NRL_stadiums /* Address data to geocode */
  out = geocoded /* Geocoded output data set */
  lookup = lookup.postcodes /* Australian lookup data */
  addresszipvar = postcode /* Address data postcode var */
  lookupzipvar = POA_code /* Lookup data postcode var */
  nocity;
run;
```

The input data preparation and geocoding program are in the sample file named ZIP_GEOCODE_AUS_930.sas. Although not included in this example, the geocoded locations were also plotted on a map of Australia with the GMAP procedure as shown in Figure 8. Note that the ABS postcode areas are also displayed as polygons.

SAS 9.3M2 – 9.4

As noted, the second maintenance release of 9.3 SAS (9.3M2) added the MAPSGFK.WORLD_CITIES data set to support international CITY geocoding. It allows PROC GEOCODE to cascade into the CITY method for any observations not located by postal code. Therefore the following syntax includes options needed if CITY geocoding is triggered:

```sas
proc geocode
  method = zip /* ZIP method for postcodes */
  data = NRL_stadiums /* Address data to geocode */
  out = geocoded /* Geocoded output data set */
  lookup = lookup.postcodes /* Australian lookup data */
  addresszipvar = postcode /* Address data postcode var */
  addresscityvar = city /* Address data city var */
  addresscountryvar = country /* Address data country var */
  lookupzipvar = POA_code; /* Lookup data postcode var */
run;
```
ADDRESSCITYVAR and ADDRESSCOUNTRYVAR are specified in the PROC GEOCODE syntax so any stadiums not located by postal code will be located by the CITY method. Since the addresses are not in the U.S., those variables are necessary if an observation cascades from ZIP to CITY geocoding.

MAPSGFK.WORLD_CITIES is the default international city lookup data set. See the GEOCODE documentation for details on non-U.S. CITY geocoding. If you wish to disable this behavior you can omit those options and add NOCITY to your PROC GEOCODE syntax.

The input data preparation and geocoding program are in the sample file named ZIP_GEOCODE_AUS_940.sas. Although not included in this example, the geocoded locations were also plotted on a map of Australia with the GMAP procedure as shown in Figure 8. Note that the ABS postcode areas are displayed as polygons.

![Postal Areas and National Rugby League Stadiums](image)

*Figure 8. Locations Geocoded by Australian Postcode*
• ZIP+4

This example uses the PLUS4 geocoding method with U.S. ZIP+4 codes. The sample address data for this example is very similar to the data for ZIP code geocoding with the addition of a ZIP+4 variable.

SAS does not ship lookup data for PLUS4 geocoding. You must either download the data from the SAS MapsOnline site or purchase data.

The example attempts to provide the location of the ZIP+4 center for each address. If a ZIP+4 is not found, then it will try to locate the ZIP code centroid. Normally if the ZIP code cannot be found, the geocoder looks for the location of the city by default, but this example will not cascaded into the CITY method because the NOCITY option is included. In addition, the attribute variable TRACT will add the Census Tract value of the geocoded location to the output data.

The procedure syntax is:

```
proc geocode
  method       = plus4          /* Geocoding Method */
  lookup       = lookup.zip4    /* Lookup data set */
  data         = work.customers /* Input address data */
  out          = work.geocoded  /* Output data set */
  attributevar = (tract)       /* Added output variable */
  nocity;                     /* Disable CITY method */
run;
```

The complete code for this program is in the ZIP4_GEOCODE_US.sas file. The sample also calls the %DELWARE_MAPS macro to plot the points on a Delaware map as in Figure 5.

• STREET

U.S. street level geocoding is supported for SAS releases 9.2M3 – 9.4 with Canadian street geocoding added in 9.4. The street method lookup data set format changed in SAS 9.4. If you download the lookup data or the TIGER2GEOCODE.sas import program from MapsOnline, get the version appropriate for your SAS release.

After geocoding, the point locations can be plotted on a map using:

• SAS/GRAPH GMAP Procedure
• SAS/GIS and the GIS Procedure
• SAS Visual Analytics Explorer
• SAS Bridge for ESRI

United States

This example uses the STREET geocoding method to geocode the address data for the Wake County Public Schools in North Carolina. The lookup data set is the sample lookup data for Wake County which is shipped in SASHELP. For geocoding in other areas, you can download nationwide U.S. street lookup data from SAS MapsOnline. The example attempts to find the street location for each school but also cascades into the ZIP and then the CITY methods if necessary. Note that depending on the version of the lookup data being used, the variable name in the ATTRIBUTE_VAR= option may have to be changed. The Census Bureau Tract variable name differs in the various annual releases of TIGER data.

The procedure syntax is:

```
proc geocode
  method       = street          /* Street method */
  data         = work.schools    /* Address data to geocode */
  out          = work.geocoded   /* Geocoded output data set */
  lookupstreet = sashelp.geoexm  /* Street method lookup data */
  attribute_var = (tractce00);   /* Attribute var to add */
run;
```

The code for this program is in the sample file named STREET_GEOCODE_US.sas.
Example output from the GMAP procedure as shown in Figure 9. The streets were plotted using PROC ANNOTATE and U.S. state road files available on SAS MapsOnline.

Figure 9. Wake County School Locations Geocoded by Street Method

Canada

Canadian street-level geocoding support was added in SAS 9.4. This example uses the STREET geocoding method to locate addresses of bed and breakfast inns in Quebec City, Canada.

The lookup data was created with free National Road Network (NRN) files from the GeoBase repository of Canadian geospatial data. Street lookup data for Canada can be downloaded directly from SAS MapsOnline. You can also use the %GEOBASE2GEOCODE macro program to import GeoBase NRN files to create street lookup data for specific provinces. Download the GEOBASE2GEOCODE.zip file from MapsOnline for the macro and documentation.

The procedure syntax for this example is:

```sas
proc geocode
  method            = street              /* Street method                   */
  data              = work.inns           /* Address data to geocode         */
  out               = work.geocoded       /* Geocoded output data set        */
  lookupstreet      = lookup.canada_m     /* Street lookup data set          */
  direct            = lookup.gcdirect_can /* English/French directions       */
  type              = lookup.gctype_can   /* Canadian street types           */
  addresscountryvar = country             /* Input data set country name var */
  addressstatevar   = province            /* Input data set state name var   */
  addresscityvar    = city                /* Input data set city name var    */
  nozip;                                  /* Disable ZIP method              */
run;
```

PROC GEOCODE: Finding Locations Outside the U.S.
The Canadian street lookup data sets were downloaded from MapsOnline. The libname LOOKUP was assigned to the location where those data sets were installed. The LOOKUPSTREET= option points to the primary Canadian street lookup data set CANADA_M.

The DIRECT= option is needed to specify the data set containing English and French direction prefixes/suffixes used in Canadian street names, e.g. SOUTHEAST and SUDEST. This data set is included in the Canadian street lookup data file on MapsOnline or generated when the %GEOBASE2GEOCODE macro imports GeoBase NRN files.

TYPE= specifies a data set of Canadian street type prefixes and suffixes, e.g. AVENUE and ALLÉE. This data set is included in the Canadian street lookup data file on MapsOnline or generated by the %GEOBASE2GEOCODE macro when GeoBase files are imported.

The ADDRESSCOUNTRYVAR= option specifies the variable in the input address data set (WORK.INNS) that includes the country identifier. This option is required for international city geocoding. Although performing street-level geocoding, it is needed here in case a particular address is not found by the street method and the street geocoding process cascades into the CITY method. Including the country identifier tells the geocoder to use the MAPSGFK.WORLD_CITIES lookup data set if CITY method geocoding is invoked.

ADDRESSSTATEVAR= specifies the variable in the input address data set (WORK.INNS) that identifies the Canadian province of the address.

The ADDRESSCITYVAR= option specifies the variable in the input address data set that includes the city name of the address to be geocoded.

The NOZIP option is included to disable ZIP matching attempts. The GeoBase National Road Network data does not include Canadian postcode values, so ZIP geocoding is not possible. If NOZIP is omitted, geocoding does run but with a warning in the SAS log that the ZIP method was disabled.

The code for this sample program is in the file named STREET_GEOCODE_CAN.sas.

Figure 10 shows geocoded locations of the bed and breakfast inns plotted on a street map of Quebec City, Canada. The map was created using PROC GMAP and the new functionality of the Java MapApplet with an OpenStreetMap background.
Figure 10. Locations in Quebec City Geocoded by Street Method

- IP ADDRESS

This example program uses the RANGE geocoding method to geocode, also called geolocate, IP addresses by city and country. There are no defaults with this method, so the names of both lookup data sets and variable must be specified.

The lookup data sets were created from free GeoLite Country and GeoLite City files downloaded from MaxMind. The SAS/GRAPH autocall macro %MAXMIND imported the city and country files to create the required range and lookup data sets. See the PROC GEOCODE documentation for macro usage.

The range data set contains ranges of IP addresses. The lookup data provides the latitude and longitude information for each range. A KEY variable links the two data sets. Initially the proper range for a specific IP address is found, and then the key value is used to access the lookup data set to find the latitude and longitude for that key.

RANGE geocoding is available in SAS 9.2 – 9.4 for IPv4 addresses. IPv6 addresses will be supported in a future release.
The procedure syntax is:

```sas
proc geocode
   method = range           /* Geocoding method         */
   data  = work.addresses   /* IP address input data set */
   out   = work.geocoded    /* Geolocated output data set*/
   addressvar = IPaddress   /* Address var in input data set */
   rangedata = lookup.cityblocks /* Range data set */
   beginrangevar = startipnum /* Begin range var in range data */
   endrangevar = endipnum   /* End range var in range data */
   rangekeyvar = locid      /* Link var in range data set */
   lookup = lookup.citylocation /* Lookup data set */
   lookupkeyvar = locid     /* Link var in lookup data set */
   lookupyvar = latitude    /* X variable */
   lookupxvar = longitude   /* Y variable */
   attributevar = (city, country); /* Additional vars wanted */
run;
```

The code for this sample program is in the file named `IP_GEOCODE.sas`. After geolocating the IP addresses, the macro `%WORLD_MAPS` can be invoked to plot their locations on a world map from the MAPS library as seen in Figure 11.

![IP Addresses](IP_GEOCODE.png)

**Figure 11. Geocoded IP Address Locations**
CUSTOM

CUSTOM geocoding is a flexible lookup method that enables you to geocode with your own geographic regions. In this example, the address data is a list of customers for which only their telephone area codes are available. The lookup data is created by extracting the necessary area code data from the SASHELP.ZIPCODE data set. Because this is a custom file, you must specify the name of the lookup data set, the name of the address variable and the name of the lookup variable.

The following code creates the custom area code lookup file. An average latitude and longitude is computed for each area code in SASHELP.ZIPCODE:

```sas
proc sort data=sashelp.zipcode out=temp;
  by areacode;
run;
proc means data=temp noprint;
  by areacode;
  label x = 'Mean Longitude of Area Code'
         y = 'Mean Latitude of Area Code'
         areacode = 'Area Code';
  var x y;
  output mean= out=work.areacodes (keep=x y areacode label='Area Code Centers');
run;
```

The geocoding procedure syntax using that custom lookup data set is:

```sas
proc geocode
  method = custom /* Geocoding method */
  data = customers /* Input data set to geocode */
  out = geocoded /* Geocoded output data set */
  lookup = work.areacodes /* Lookup data set */
  lookupvar = areacode /* Geocode var in lookup data */
  addressvar = areacode; /* Customer var to geocode */
run;
```

The CUSTOM method can be applied to almost any data which has a geographic component and contains a common key with the data to be geocoded.

The complete code for this example is in the file CUSTOM_GEOCODE.sas.

SUMMARY

PROC GEOCODE is included in the SAS/GRAPH product. It computes geographic locations for several forms of address data which can then be plotted on a map or used as input for spatial analysis. The geocoder can also assign attribute data values to the geocoded locations.

The procedure is both simple to use and extremely flexible by allowing geocoding with various location-related parameters. SAS provides free lookup data for several geocoding methods plus the ability to import data from various spatial data providers and government agencies.
RESOURCES

SAS Global Forum and SUGI papers and presentations on geocoding and mapping, with example programs, include:

- “PROC GEOCODE: Now with Street-Level Geocoding,” SAS Global Forum 2010
- “PROC GEOCODE: Creating Map Locations from Your Data,” SAS Global Forum 2009
- “Tips and Tricks III: More Unique SAS/GRAPH Maps.” SUGI 30
- “Tips and Tricks II: Getting the most from your SAS/GRAPH maps.” SUGI 29

SAS Global Forum Proceedings

SAS Technical Support

SAS/GRAPH Documentation with PROC GEOCODE and GMAP

SAS/GIS Documentation

SAS Bridge for ESRI

SAS MapsOnline

CONTACT INFORMATION

Your comments and questions are valued. Contact the authors at:

Darrell Massengill
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Darrell.Massengill@sas.com

Ed Odom
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Ed.Odom@sas.com

Assistance with the example programs is available through SAS Technical Support.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

Other brand and product names are trademarks of their respective companies.
APPENDIX 1 - ZIP CODE FREQUENTLY ASKED QUESTIONS

What is a ZIP code?
It is a five-digit number identifying a U.S. Postal Service delivery route or location. ZIP codes were developed in the 1960s to speed mail sorting.

What does a ZIP code represent?
Most ZIP codes are usually a group of linear delivery routes (streets). However, a ZIP code can also be a single location, such as a post office, a large commercial building, a university, a military base, or even a navy ship.

Do ZIP codes change?
Yes. ZIP codes change as the numbers of customers on mail carrier routes increase with population changes.

Do ZIP codes cross county or state lines or Census Blocks or Tracts?
Yes.

Does SAS provide ZIP code data?
Yes. The SASHELP.ZIPCODE data set lists each ZIP with its centroid (latitude and longitude), city, state, county, telephone area code, Metropolitan Statistical Area (MSA) and time zone.

How often is SASHELP.ZIPCODE updated?
The most current version is included in each SAS release. Normally, only one or two new ZIP codes are added per update.

Can I get more frequent updates of SASHELP.ZIPCODE?
Yes. The data set is updated quarterly and available from the SAS Maps Online Web site.

Can I get older ZIP code data for a historical analysis?
Yes. You can download previous versions of SASHELP.ZIPCODE from the SAS Maps Online Web site.

Does SAS provide ZIP code boundary maps?
No. ZIP codes represent general delivery routes and points, not polygons. Creating seamless polygons to enclose these is very time-consuming and labor-intensive. Because ZIP codes also change, they have to be redone on a regular basis.

Does the Postal Service provide ZIP code boundary maps?
No.

Who does provide ZIP code boundary maps?
Various third-party spatial data vendors do. They can easily be located with an Internet search. Because the Postal Service does not provide these maps, there are no official ZIP code boundaries. Because each data vendor generates these using different methods, there will likely be some differences among them.

Why are ZIP code boundary maps so expensive?
See the previous answer explaining why SAS does not provide these maps.

What is ZIP+4?
ZIP+4 is an optional four-digit add-on to the standard ZIP code. It basically subdivides a larger ZIP code delivery area into smaller carrier routes. It is used primarily by businesses or large commercial mailers to pre-sort mail for a bulk discount. Figure 2 illustrates how a single street in a ZIP code is assigned a ZIP+4 value.

Does SAS provide ZIP+4 data?
Yes but be aware that as of 2013, the ZIP+4 lookup data on MapsOnline was generated from Census Bureau 2006 Second Edition TIGER files. It includes ZIP code, ZIP+4, latitude, longitude, street name, Census Tract and Block, city, state/county FIPS code, and Census Feature Class Code (CFCC) values. Because this data set is too large to ship with SAS, you can download it from the SAS Maps Online Web site. It is in SAS transport format and includes a SAS CIMPORT program to convert it into a SAS data set for your platform. This data set can be used with the PLUS4 method in PROC GEOCODE for ZIP+4 geocoding.
Who else provides ZIP+4 data?
Various third-party data vendors do. For example, the Melissa Data GEO*Data product contains ZIP+4 centroids. SAS provides the %GCDMET9 autocall macro to import these files into SAS data sets for use by PROC GEOCODE.

How often is the SAS ZIP4 data set updated?
The last release from the Census Bureau that contained ZIP+4 values was the 2006 Second Edition TIGER/Line files. For the 2008 TIGER release, the Census Bureau changed from their Record Type (RT) format to the shapefile format (.shp) and did not include ZIP+4 values. They do intend to restore them to TIGER data, but as recently as the 2012 TIGER files the ZIP+4 values were still missing. We will update the ZIP4 lookup data on MapsOnline when the Census Bureau restores those values to their TIGER data.

Does SAS provide ZIP+4 boundary maps?
No. See the previous answer explaining why SAS does not provide ZIP code maps.

What are ZCTAs?
ZIP Code Tabulation Areas (ZCTA) are generalized ZIP code maps created by the Census Bureau. It says that a ZCTA is “a statistical geographic entity that approximates the delivery area for a U.S. Postal Service five-digit or three-digit ZIP code.” ZCTAs are closed polygons. Not all ZIP codes are represented by a ZCTA polygon because, as noted earlier, some ZIP codes represent discrete points, not areas that can be enclosed by a polygon.

Where can I get ZCTA data?
It is available by state in several formats from the U.S. Census Bureau.

How can I use ZCTA data in SAS?
The SAS/GRAPH MAPIMPORT Procedure imports ZCTA shapefiles (.shp) into SAS/GRAPH map data sets and computes the ZCTA polygon centroids. SAS/GIS can import the ARC/INFO Export (.e00) format into SAS/GIS as a polygonal map layer and also compute the centroids.

How often are ZCTAs updated?
ZCTAs were originally released using 2000 Census data and were updated for the 2010 Census.

Where can I get additional ZCTA information?
The Census Bureau maintains a ZCTA FAQ.

What about postal code data for other countries?
SAS provides macro programs which will import postal code files for Great Britain (England and Scotland) and Australia to generate lookup data for use with PROC GEOCODE. Look on the geocoding download page on SAS MapsOnline for the CODEPOINT2GEOCODE.zip and ABS2GEOCODE.zip files.

These zipped files include the macro programs with usage instructions as well as information on how to acquire the postal data files.

Data vendor ZIP Code Download includes Canadian postcode centroid locations in their Premium Edition product. SAS MapsOnline contains the %ZIPCODEDOWNLOAD2GEOCODE macro program which imports the Premium Edition CSV file to create Canadian postal code lookup data.

Other third-party spatial data vendors supply postal code data for various countries. One example is the MapMechanics site. Additional postal code data vendors can be located with an Internet search. You can use PROC IMPORT, PROC MAPIMPORT or the DATA step to import it for use as PROC GEOCODE lookup data. Contact SAS Technical Support for assistance with a specific format.
APPENDIX 2 - SAS SPATIAL CAPABILITY SUMMARY

DISPLAY ON MAP

Geocoding assigns latitude and longitude values to your addresses. Those geocoded locations can then be displayed on a map:
- Annotate them onto a SAS/GRAPH map
- Import them as a point layer into a SAS/GIS map
- Create a Geo Role in SAS Visual Analytics Explorer (VAE)
- Use the SAS Bridge for ESRI to display them in ArcView
- Import them into a third-party mapping system, for example, Google Maps

BASE SAS FUNCTIONS

SAS address- or coordinate-related functions:
- FIPNAME - Convert two-digit FIPS code to uppercase state name
- FIPNAMEL - Convert two-digit FIPS code to mixed case state name
- FIPSTATE - Convert two-digit FIPS code to two-character state postal code
- GEODIST - Calculate geodetic distance between latitude and longitude coordinates
- STFIPS - Convert state postal code to FIPS state code
- STNAME - Convert state postal code to uppercase state name
- STNAMEL - Convert state postal code to mixed case state name
- ZIPCITY - Return city name and two-character postal code for a ZIP Code
- ZIPCITYDISTANCE - Calculate geodetic distance between ZIP Codes
- ZIPFIPS - Convert ZIP Code to two-digit FIPS code
- ZIPNAME - Convert ZIP Code to uppercase state name
- ZIPNAMEL - Convert ZIP Code to mixed case state name
- ZIPSTATE - Convert ZIP Code to two-character state postal code

SPATIAL ANALYSIS

Spatial analysis involves determining the effect of position on response data. Geocoded X and Y locations can be used as the basis for spatial analyses. Several SAS procedures and products perform spatial analysis of attribute values associated with X and Y locations and compute predicted values at other positions:
- PROC VARIOGRAM
- PROC KRIGE2D
- JMP