### PROC GEOCODE Examples

**Example 1: Geocode by ZIP code**

You have a data set of customers and their ZIP codes and want to find their locations. This example uses default lookup data set SASHELP.ZIPCODE.

```sas
proc geocode method=zip data=customers out=geocoded; run;
```

**Example 2: Geocode by CITY and STATE**

You have a data set of cities and U.S. states and want to find their locations. This method uses the default lookup data set MAPSGFK.USCITY_ALL.

```sas
proc geocode method=city data=suppliers out=geocoded; run;
```

**Example 3: Assign associated data to address**

Lookup data can contain additional information that can be assigned to geocoded locations. Use Example 1 to geocode customers by ZIP code and also assign state and county names to the geocoded points.

```sas
proc geocode method=zip data=customers out=geocoded attribute_var=(statename countynm); run;
```

**Example 4: Geocode by British postal code**

To locate stores in Great Britain by postal codes, download the CodePoint2Geocode.zip file from SAS MapsOnline. Then download and import postal code lookup data from the British Ordnance Survey for use as zip method lookup data.

```sas
proc geocode method=zip data=stores out=geocoded; run;
```

### PROC GEOCODE Examples (Continued)

**Example 5: Geocode by street and house number**

You have a data set of delivery addresses and want to locate them. Specify the libref to the street lookup data downloaded from MapsOnline, `specify street geocoding method, 3 input data set of addresses to locate, 4 output data set, and 5 US street lookup data from MapsOnline.

```sas
libname usdata 'SAS-Jlibrary';
proc geocode method=street data=deliveries out=geocoded lookupstreet=usdata.usm; run;
```

**Example 6: Geocode IP addresses**

You have IP addresses from a web site and wish to find their locations. The IP address lookup data must be downloaded from MaxMind.com and imported with the %MaxMind autocall macro into a SAS data set. Specify the libref to that IP lookup data set location, 2 use range method, 3 input data set, 4 output data set, 5 input data set variable with dotted-quad IP addresses to locate, 6 lookup data set of IP address ranges, 7 variable with starting IP address range values, 8 variable with ending IP address range values, 9 variable linking into lookup data set, 10 lookup data set with coordinates, 11 variable with the link from the range data set, 12 longitude/latitude variables in the lookup data set, and 13 variables from the lookup dataset to assign to the IP address locations.

```sas
libname ipdata 'SAS-Jlibrary';
proc geocode method=range data=ipdata.rangevars out=geocoded addressvar=ip_dottedqua startipvar=startip endipvar=endip lookup=ipdata.cityblocks lookupcityvar=locid lookupipvar=iplookup attributevar=(country region city countryname); run;
```

### PROC GEOCODE Tips

- Each geocoding method requires specific lookup data. Some data sets are shipped in SASHELP and MAPSGFK, some downloaded from SAS MapsOnline (support.sas.com/mapsonline), and others are available from third party providers.

- **STREET method geocoding is the most precise.** It uses a house number to place an address along the specified street in its city and state or ZIP code.

- The ZIP method assigns the latitude and longitude of the center of an address’ ZIP code.

- The CITY method provides the center of the address’ city and state or city and country.

- The PLUS4 method returns the center of the ZIP+4 for each address.

- You can specify any of the above methods directly. Some will also cascade to a lower precision method for any address which is not matched. For example, if the STREET method fails to match an address, the ZIP method is tried. If that fails, the CITY method is attempted.

- The RANGE method determines IP address locations.

- The CUSTOM method provides a way to use your own geocoding lookup data.

For complete information, refer to the SAS 9.4 documentation at support.sas.com/documentation.

### SAS 9.4 GEOCODE Procedure Tip Sheet

This tip sheet places frequently used information on one sheet of paper. It covers the SAS GEOCODE procedure.

PROC GEOCODE was part of SAS/GRAPH through 9.4M4 release. In 9.4M5 it moved into Base SAS and also became available in the SAS University Edition.

The GEOCODE procedure determines locations from addresses. It can also assign information associated with those locations to the geocoded points.

PROC GEOCODE determines the latitude and longitude of an address at the street level, ZIP code, ZIP+4 or city depending on the precision needed. It can also assign attribute values to those locations such as Census Tract, Block and Block Group.

PROC GEOCODE also supports IP address geolocation. Its custom geocoding method allows geocoding using specialized geographic areas such as sales regions or telephone area codes.

The resulting latitude and longitude can be plotted on a map, used in distance computations or in spatial analytics.
### PROC GEOCODE Syntax

```sas
PROC GEOCODE
  <ADDRESSCITYVAR=character-variable>
  <ADDRESSCOUNTRYVAR=variables>1
  <ADDRESSPLUS4VAR=variable>
  <ADDRESSSTATEVAR=character-variable>
  <ADDRESSVAR=variable>
  <ADDRESSZIPVAR=variable>
  <ATTRIBUTEVAR=variable 1,...  variable-n>
  <BEGINRANGEFVAR=variable>
  <DATAT=inputs-address-data-set>
  <DIRECT=street-direction-data-set>1
  <ENDRANGEVAR=variable>
  <FIPS=FIPS-code-data-set>3
  <LOOKUP=lookup-data-set>
  <LOOKUPDATA=lookup-data-set>1
  <LOOKUPCITYVAR=character-variable>
  <LOOKUPCOUNTRYVAR=variable>1
  <LOOKUKKEYVAR=variable>
  <LOOKUPLONGVAR=variable>
  <LOOKUPPLUS4VAR=variable>
  <LOOKUPSTATEVAR=character-variable>
  <LOOKUPVAR=variable>
  <LOOKUPZIPVAR=variable>
  <METHOD=geocoding-method>
  <NOZIP>
  <NOSTIM>
  <NOCITY>
  <OUT=geocoded-output-data-set>
  <RANGEDATA=IP-address-lookup-data-set>
  <RANGEDECIMAL>
  <RANGEKEYVAR=variable>
  <STATE=state/province-data-set>1
  <TYPE=street-type-data-set>
```

### PROC GEOCODE Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ADDRESSCITYVAR=</code></td>
<td>specifies the character variable in input address data set containing city names</td>
</tr>
<tr>
<td><code>ADDRESSCOUNTRYVAR=variables</code></td>
<td>specifies the character variable in input address data set containing country names or identifiers</td>
</tr>
<tr>
<td><code>ADDRESSPLUS4VAR=variable</code></td>
<td>specifies the variable in input address data set containing ZIP+4 extensions</td>
</tr>
<tr>
<td><code>ADDRESSSTATEVAR=character-variable</code></td>
<td>specifies character variable in input address data set containing two-character state abbreviations</td>
</tr>
<tr>
<td><code>ADDRESSVAR=variable</code></td>
<td>specifies the address variable for STREET, CUSTOM or RANGE geocoding</td>
</tr>
<tr>
<td><code>ADDRESSZIPVAR=variable</code></td>
<td>specifies the variable in the input address data set containing 5-digit ZIP codes or non-US postcodes</td>
</tr>
<tr>
<td><code>ATTRIBUTEVAR=variable 1,... variable-n</code></td>
<td>lists non-geocoding variables in lookup data to be added to the output data set for geocoded locations</td>
</tr>
<tr>
<td><code>BEGINRANGEVAR=</code></td>
<td>specifies an alternate data set of street directions (default is SASHELP.GCDIRECT)2</td>
</tr>
<tr>
<td><code>DATAT=inputs-address-data-set</code></td>
<td>specifies the numeric variable in range data set with beginning IP address for each range</td>
</tr>
<tr>
<td><code>DIRECT=street-direction-data-set</code></td>
<td>specifies the SAS data set containing address observations to be geocoded</td>
</tr>
<tr>
<td><code>ENDRANGEVAR=</code></td>
<td>specifies the numeric variable in range data set with ending IP address for each range</td>
</tr>
<tr>
<td><code>FIPS=FIPS-code-data-set</code></td>
<td>specifies option to specify alternate data set to convert FIPS codes to state names</td>
</tr>
<tr>
<td><code>LOOKUP=</code></td>
<td>specifies the lookup data set associating coordinates with addresses for specific geocoding methods</td>
</tr>
<tr>
<td><code>LOOKUPCITY=</code></td>
<td>specifies the lookup data set for the CITY method1</td>
</tr>
<tr>
<td><code>LOOKUPCITYVAR=</code></td>
<td>specifies the character variable in lookup data set containing city names</td>
</tr>
<tr>
<td><code>LOOKUPCOUNTRYVAR=</code></td>
<td>specifies the character variable in lookup data set containing country names or identifiers</td>
</tr>
<tr>
<td><code>LOOKUPKEYVAR=</code></td>
<td>specifies variable in lookup data set that links to the RANGECITYVAR variable in RANGEDATA= data set</td>
</tr>
<tr>
<td><code>LOOKUPPLUS4VAR=</code></td>
<td>specifies the numeric variable in the lookup data set containing longitude values3</td>
</tr>
<tr>
<td><code>LOOKUPSTATEVAR=</code></td>
<td>specifies the character variable in the lookup data set containing latitude values3</td>
</tr>
<tr>
<td><code>LOOKUPZIPVAR=</code></td>
<td>specifies the numeric variable in the lookup data set containing 5-digit ZIP codes or non-US postcodes</td>
</tr>
<tr>
<td><code>LOOKUPDIRECTVAR=</code></td>
<td>specifies character variable in input address data set containing two-character state abbreviations</td>
</tr>
<tr>
<td><code>LOOKUPVART=primary street lookup data set for STREET geocoding method</code></td>
<td>specifies the primary street lookup data set for STREET geocoding method</td>
</tr>
<tr>
<td><code>STATE=state/province-data-set</code></td>
<td>specifies the variable in lookup data set containing non-address values for CUSTOM geocoding</td>
</tr>
<tr>
<td><code>TYPE=street-type-data-set</code></td>
<td>specifies an alternate street data set to use in the STREET geocoding method to standardize common street address elements (default is SASHELP.GCTYPE)2</td>
</tr>
</tbody>
</table>

1 Added in SAS 9.3M2
2 Added in SAS 9.4
3 Deprecated in SAS 9.4

---

### Notes

- **METHOD=** specifies the ZIP, CITY, STREET, PLUS4, RANGE or CUSTOM geocoding method.
- **NOCT=** disables secondary matching attempt by city and state if STREET or ZIP method geocoding does not find a match.
- **NOSTIM=** disables informational messages in the SAS log that track progress of the geocoding process.
- **NOZIP=** disables secondary matching attempt by ZIP code if PLUS4 or STREET geocoding does not succeed.
- **OUT=** specifies the data set to write geocoded addresses.
- **RANGEDATA=** specifies the data set that associates ranges of IP addresses with locations.
- **RANGEDECIMAL=** notes that values of the ADDRESSVAR= variable are decimal IP addresses instead of dotted-quad.
- **RANGEKEYVAR=** specifies the variable in the range data set that links to the LOOKUPKEYVAR= variable in the LOOKUP= data set.
- **STATE=** specifies alternate data set used in STREET geocoding to equate state/province abbreviations and full names (default is SASHELP.GCSTATE)2.
- **TYPE=** specifies an alternate SAS data set to use in the STREET geocoding method to standardize common street address elements (default is SASHELP.GCTYPE)2.

---

### Added in SAS 9.3M2

1 Added in SAS 9.3M2
2 Added in SAS 9.4
3 Deprecated in SAS 9.4

---

(Continued)