pitney bowes

Spectrum™ Technology Platform
Version 12.0

GeoComplete Guide

Table of Contents

1 - Enterprise GeoComplete

2 - API

1 - Enterprise GeoComplete

In this section

GeoComplete Module

The GeoComplete module for Spectrum™ Technology Platform automatically suggests addresses and Points of Interest (POIs) as you type and immediately returns candidates based on your input. This "as you type" recognition of addresses and points of interest is particularly valuable for mobile applications where you can pinpoint candidates on a map. GeoComplete can be incorporated into many types of business applications.

> **Note:** POI data is provided in separate data bundles and must be separately purchased. POI and category/subcategory features are available only if you have licensed and installed POI data.

The GeoComplete component is delivered as a Spectrum™ Technology Platform module and can be built into stages just like any other component. Spectrum™ Technology Platform must already be installed.

GeoComplete Database Resources

The GeoComplete Module databases are installed on the Spectrum™ Technology Platform server. The databases are available by subscription from Pitney Bowes and are updated regularly.

The GeoComplete databases are delivered in several zip files according to region. Installing one zip file installs all the included country databases. However, each country is separately licensed. See updated Release Notes for changes and information on newly supported countries.

Adding a Database Resource

To create a GeoComplete database resource:

1. If you haven't already done so, install the database files on your system. For instructions on installing databases, see the *Spectrum™ Technology Platform Installation Guide*.
2. In Management Console, under **Resources**, choose **Spectrum Databases**.
3. Click the Add button ⊞ to create a new database or select an existing database resource then click the Edit button ✎ to change it. You can also create a new database resource by copying an existing one, by clicking the Copy button ⊞.
4. If you are creating a new database, enter a name for the database resource in the **Name** field. The name can be anything you choose. If you are creating a new database by copying an existing

one, rename the default name as needed. You cannot modify the name of an existing database resource, because any services or jobs that reference the database resource by its original name would fail.

5. In the **Pool size** field, specify the maximum number of concurrent requests you want this database to handle.

   The optimal pool size varies by module. You will generally see the best results by setting the pool size between one-half to twice the number of CPUs on the server, with the optimal pool size for most modules being the same as the number of CPUs. For example, if your server has four CPUs you may want to experiment with a pool size between 2 (one-half the number of CPUs) and 8 (twice the number of CPUs) with the optimal size possibly being 4 (the number of CPUs).

   When modifying the pool size you must also consider the number of runtime instances specified in the dataflow for the stages accessing the database. Consider for example a dataflow that has a Geocode US Address stage that is configured to use one runtime instance. If you set the pool size for the US geocoding database to four, you will not see a performance improvement because there would be only one runtime instance and therefore there would only be one request at a time to the database. However, if you were to increase the number of runtime instances of Geocode US Address to four, you might then see an improvement in performance since there would be four instances of Geocode US Address accessing the database simultaneously, therefore using the full pool.

   Tip: You should conduct performance tests with various settings to identify the optimal pool size and runtime instance settings for your environment.

6. In the **Module** field, select InternationalGeocoderGeoComplete.

7. In the **Type** field, select GeoComplete.

8. Click the Add button ⊞ to install a new dataset. In the **Name** field, specify a name for this dataset. This name can be anything you choose. In the **Path** field specify the folder that contains the datasets for the country you selected. You can type in a partial path and click the Browse button 📄 to drill down into the file structure to locate the datasets you need. When you click OK, the dataset for the country will be visible in the database resource list on the Add Database page.

   The datasets are available in bundles: AMER, APAC and EMEA. The default installation location is: `<InstallLocation\AMER\data>` where <InstallLocation> is the directory you specified when you installed the datasets. Only specify the location of the datasets you have licensed.

   > **Note:** Do not put datasets on a network drive. Doing so can cause performance problems.

9. If you have additional datasests to add, click the Add button ⊞, otherwise click **Save**.

10. Click **OK** to leave the Add Database page.

11. If there are any open Enterprise Designer sessions, click the refresh button to see the new stage.

GeoComplete Coverage

The GeoComplete Module covers street addresses for the following countries.

Note: As of the Spectrum™ Technology Platform 10.1 release, POI data must be separately purchased. POI and category/subcategory features are available only if you have licensed and installed POI data. Street address data is packaged with the GeoComplete module.

Country Name

Andorra (AND)

Australia (AUS)

Austria (AUT)

Bahrain (BHR)

Belgium (BEL)

Brazil (BRA)

Canada (CAN)

Czech Republic (CZE)

Denmark (DNK)

Finland (FIN)

France (FRA)

Germany (DEU)

Great Britain (GBR)

Hungary (HUN)

Ireland (IRL)

Country Name

Italy (ITA)

Japan (JPN)

Kuwait (KWT)

Liechtenstein (LIE)

Luxembourg (LUX)

Mexico (MEX)

Netherlands (NLD)

Norway (NOR)

Oman (OMN)

Poland (POL)

Portugal (PRT)

Qatar (QAT)

Russia (RUS)

Saudi Arabia (SAU)

Singapore (SGP)

Slovakia (SVK)

Slovenia (SVN)

South Africa (ZAF)

Spain (ESP)

Sweden (SWE)

Country Name
Switzerland (CHE)
Thailand (THA)
Turkey (TUR)
United Arab Emirates (ARE)
United States of America (USA)

Note:

See your current Spectrum™ Technology Platform Database Release Notes for more details of country coverage and data vintages.

GeoComplete Module Features

The GeoComplete Module provides the following capabilities when built into your Spectrum™ Technology Platform application. The following capabilities are supported.

- Select country (optional). See **GeoComplete Coverage** on page 6 for a list of available countries.
- Single and multiple line input search for street addresses
- Candidates returned in order based on distance from X/Y origin
- Search points of interest (POI)
- Search All to include both street addresses and POI
- Search POI categories and subcategories
- Localized categories and subcategories for countries with localized data.
- Search for brand (such as a specific hotel chain) in categories and subcategories
- Search within a bounding box
- Fuzzy match capabilities
- Matched fields reported with returned candidates

Note: POI and category/subcategory features are available only if you have licensed and installed POI data.

For both address searches and POI searches, candidates are displayed as you type. As you type more specific information, the results are refined to display more relevant candidates. Candidates

include the full address, POI category and subcategory (if the candidate is a Point of Interest), and indicate the distance from the origin.

The GeoComplete Module is packaged with a sample application that demonstrates many capabilities. This sample application includes a mapping feature, but it does not represent all the capabilities of GeoComplete. This is for sample and demonstration purposes only and is not a supported application. To use the sample application, run:

```
http://<serverIP>:8080/geocomplete
```

After installing and deploying the GeoComplete module, you can use the Management Console to explore all the capabilities and see candidate results. You can also include GeoComplete as a stage in your dataflow in Enterprise Designer.

You can expose additional features and preferences through the REST and SOAP APIs. See **Using the APIs** on page 25.

Point of Interest Features

With Point of Interest (POI) searching, you can enter a specific name (such as a hotel name), a category (such as Restaurant), or subcategory (such as Thai Restaurant).

> **Note:** POI and category/subcategory features are available only if you have licensed and installed POI data.

The entry format is either of the following:

```
<brand name, category, or subcategory> <optional area>
```

```
<optional area> <brand name, category, or subcategory>
```

where:

`<brand name, category, or subcategory>` is a point of interest, such as the name of a specific bank, restaurant, or hospital, or a category (such as parking lot, or petrol station), or subcategory (such as Thai Restaurant).

`<optional area>` can be a street name, city, state/province, postcode or any combination of searchable address fields. This area information is optional, but if provided, it will limit the candidates to the specified area.

As you type more specific information, the results are refined to display more relevant candidates. Candidates include the full address, POI category and subcategory, and indicate the distance from the origin. See **Returned Candidate Fields** on page 20 for a list of additional fields returned with POIs.

Instead of providing an area for POI search, you can constrain results by the search extent from the origin or by a bounding box.

For examples that illustrate POI , see **Typical Application Examples** on page 22.

Point of Interest Categories and Subcategories

Category and Subcategory Classification

Points of Interest (POI) are classified into categories and subcategories. These categories/subcategories are returned with POI candidates and you can also use these categories/subcategories for searching.

This two-level classification is simple and allows you to search and identify POIs including, but not limited to, restaurants, hotels, entertainment sites, hospital/medical facilities, shopping and retail establishments, and other businesses. The category and subcategory names are appropriate for terminology variations. For example, Cash Point, Cash Dispenser, Cash, and ATM are all terms that identify cash dispensing devices. You can search any of these terms with the GeoComplete Module to identify cash dispensing devices.

Categories are broad in nature, such as Restaurants. Subcategories are more specific, such as Barbeque restaurants, Barbeque, or BBQ. Any other type of ethnic or specialty restaurant could also be a subcategory. Similarly, Automotive is a category with many subcategories, including Car service, Car repair, Car sales, Tire services/tyre services (American and British spelling variations of tire and tyre).

> **Note:** POI and category/subcategory features are available only if you have licensed and installed POI data.

Localization of Categories and Subcategories

For many countries, you can search on categories and subcategories in one or more supported languages in addition to English. This can return categories and subcategories in the same language as the input. For most countries, the behavior is as follows:

- If the category is searched in the localized language, the candidate is returned with localized category name and English language subcategory name.
- If the subcategory is searched in the localized language, the candidate is returned with localized subcategory name and English language category name.

For example, Italian is a supported language in both Italy (ITA) and Switzerland (CHE). If you search for subcategory "ristorante de pesce", suggestions for seafood restaurants will be returned with the localized subcategory name.

For countries in which the data is fully localized, both the category and subcategory can be returned in the localized input language.

New countries and localized category/subcategory support will be described in regularly updated release notes.

POI Name Searching

Points of Interest can also be searched by using the name of a business, brand, institution, or other specifically named site. These names are also are returned with POI candidates, if available.

For example, you can search for a specific coffee shop, sports stadium, hospital, hotel, park, shop, restaurant, tourist attraction, or any other site. This may be important if you are looking for a specific brand name hotel/motel chain or cafe in your location.

> **Note:** POI and category/subcategory features are available only if you have licensed and installed POI data.

See **Typical Application Examples** on page 22 for examples of POI searching with the GeoComplete Module.

FindNearest Point of Interest

The Find Nearest Points of Interest feature finds the Points of Interests (POIs) nearest to a given location. The FindNearestPOIs method is in the AutoSuggestAPI class:

```
com.mapinfo.mapmarker.autosuggest.api.AutoSuggestAPI.findNearestPOIs
```

> **Note:** POI features (including FindNearest Point of Interest) are available only if you have licensed and installed POI data. FindNearestPOI will not return any candidates unless you have licensed and installed POI data.

The FindNearestPOIs method accepts the following parameters and constraints.

Table 1: FindNearestPOIs Parameters and Constraints

Parameter or Constraint	Values	Required or Optional and Usage
Country	`CTY` The 3-character country ISO code	Optional. If not specified, all available countries are searched.
Origin	The latitude/longitude of the origin. This defines the center of the search area.	Required.
Maximum number of candidates	`MaxCandidates` The maximum number of POI candidates returned. The maximum is 100. The default is 5.	Optional. If maximum number of candidates is not specified, all the POIs within the Search Distance are returned. If not specified, the default of 5 candidates are returned.

Parameter or Constraint	Values	Required or Optional and Usage
Search Distance	`SearchDistance`: The maximum distance from the origin to search.	Optional. If search distance is not specified, the search distance is 5.
Search Distance Units	`Units`: The search distance measurement unit. This can be kilometers, miles, meters, or feet. The default is kilometers.	Optional. If not specified, the default distance unit is kilometers
Locale	Locale is enabled only for FindNearest POI. English is always available and that is the default locale. You can also select Native if the POI categories and subcategories have been localized. Native refers to the local language for that country.	Optional. If the locale is specified and that locale is available, then POIs are returned in that language. If the locale is specified but that locale is not available, then POIs are returned in English. If the locale is **not** specified, POIs are returned in the native (non-English) language, if available; otherwise POIs are returned in English.

Find Nearest POI in Management Console

To use this feature from the Management Console:

1. From the **GeocComplete** tab under **Services** > **Enterprise Geocomplete Module** > **GeoComplete**, select **FindNearest POI** . The enables the Locale selection.
2. From the Locale box, select the preferred locale. English is the default but you can select Native for localized categories/subcategories. Localized content is available for selected countries only.
3. Select Max Candidates (default is 5) and Distance Unit (default is kilometers).
4. From the **Search Restrictions** tab, select the Search Distance (default is 0). Zero means that there is no search distance restriction.
5. To find the nearest point of interest, provide a name or brand and the search area origin (longitude/latitude) in the **Preview** tab and click **Run Preview**.

Street Searching Features

With street searching you can search on all address elements including street name, city, locality, state/province, postcode. Auto suggest will only be performed on the main address input field whereas the remaining fields will be used to restrict candidates returned based on what is in those fields (for example, city or postal code).

The entry format can include any combination of the following address elements. Consider listing the street name before the area names for more efficient searching.

```
street_name, area_name_4,area_name_3,area_name_2,area_name_1,post_code
```

where:

`street_name, area_name_4,area_name_3,area_name_2,area_name_1,post_code` are street name, locality, city/town, county/region, state/province, and postal code, or equivalent address elements.

Fuzzy Matching Features

GeoComplete implements algorithms that optimize the retrieval of addresses and POIs, even when the input spelling is incorrect or incomplete. These capabilities are referred to as Fuzzy Match, and are implemented through match setting constraints and are exposed through the API.

The entry format is:

`<name, category, or brand name> <optional area>`

`<optional area> <name, category, or brand name>`

where:

`<name, category, or brand name>` is a point of interest, such as bank, parking lot, or petrol (gas) station.

`<optional area>` can be a city, state/province, postcode or any combination of these. Area is optional, but if provided, it will limit the candidates to the specified area.

Table 2: Fuzzy Matching Methods and Descriptions

Fuzzy Matching Method	Description	API Method	Management Console Preference
Edit Distance	This is a calculation of the minimum number of character substitutions, insertions, deletions, or transpositions required to change one word into another. This is a mathematical means of determining the similarity of words. Edit Distance is the only algorithm supported in the initial implementation of fuzzy matching, and therefore the only technique described in this document.	FuzzyMatchMode	Hard - allows for one character substitution, insertion, deletion or transposition. Soft - allows for two character substitutions, insertions, deletions or transpositions.

Fuzzy Matching by Edit Distance

Edit Distance Algorithm

The Edit Distance algorithm allows for a certain number character substitutions, insertions, deletions or transpositions per address field when attempting to match input with intended strings.

Edit Distance API

GeoComplete provides API methods to implement and control the Edit Distance fuzzy matching capabilities.

Table 3: Edit Distance API Methods

Name of API Method	Description
`FuzzyMatchMode.None`	Enable or disable Edit Distance fuzzy matching. Fuzzy Match is disabled by default.
`FuzzyMatchMode.Hard_Match`	The Hard distance match allows 1 edit per address field. At least 2 characters must match and the first character of both strings must be the same.
`FuzzyMatchMode.Soft_Match`	The Soft distance match allows 2 edits per address field. At least 2 characters must match. The first character of both strings can be different.

Fuzzy Match is disabled by default (FuzzyMatchMode.None). You can enable Hard Match, Soft Match, or disable Fuzzy Match through the Management Console Search Preferences.

Fuzzy Match Options in the Management Console

GeoComplete provides preferences in the Spectrum™ Technology Platform Management Console to implement the fuzzy matching capabilities.

When you select Search Type of GeoComplete Address, GeoComplete Category, GeoComplete POI, or GeoComplete All, the Fuzzy Match Settings are enabled. From this selection box you can choose None (to disable Fuzzy Match), Soft Match, or Hard Match.

Multiline Support

The GeoComplete Module supports input values in a single field or in multiple fields. When partial information in AddressLine1 is combined with other fields, such as city or postal code, the returned candidates are further refined than if only the address field was used.

Management Console provides a preview to demonstrate the use of multiple fields in a search. Multiple fields can also be used in Enterprise Designer as a way to enhance a GeoComplete flow where candidates are refined before the geocoding process begins. The APIs also provide the methods to construct REST and SOAP requests using multiple fields.

Multiple Countries and Dictionaries

If you use GeoComplete with more than one country selected, rules are used to determine which country has search priority. You can also have more than one dictionary per country to support standard and custom user dictionaries. For some countries, multiple dictionaries can support more than one language.

Multiple Country Support and Search Priority

If multiple countries are available to GeoComplete, countries are searched based on default rules and user input. The priority rules are applied in the following order:

1. If a country code is provided then search is done within this country.
2. If a country code is absent but the point location is provided, the location is used to determine the country. You can also use a bounding box instead of a reference location to define a search area. If the bounding box overlaps more than one country, you can potentially get candidates from more than one country. If one of these is the default country, candidates will come from that country. If none of the countries intercepted by the bounding box is the default country, then all of the intercepted countries are searched in the order/priority in which the country data was installed, until the requested number of candidates are found.
3. If the country code is not specified and neither the location or bounding box are given, then the default country is used.
4. If the country code is not specified and neither the location or bounding box are given and no default country is provided, then all available countries are searched in the order/priority in which the country data was installed until the requested number of candidates are found.

When GeoComplete searches all loaded countries, significant performance issues may be encountered. If your application calls for searching all loaded GeoComplete countries, it is advisable to constrain the results using Management Console.

Language Support

Some countries can include multiple dictionaries to support more than one language. When you install databases for the following countries, you can select more than one language dictionary:

Table 4: GeoComplete Countries with Multiple Language Support

Country	Language Databases
Bahrain (BHR)	English and Arabic
Japan (JPN)	English and Japanese
Kuwait (KWT)	English and Arabic
Oman (OMN)	English and Arabic
Qatar (QAT)	English and Arabic
Russia (RUS)	English and Russian
Saudi Arabia (SAU)	English and Arabic
United Arab Emirates (ARE)	English and Arabic

Candidates are ordered by how close they are to the origin regardless of language. However, if candidates are equidistant from the origin, then candidates matching the input language are listed first.

GeoComplete Options

You can use several GeoComplete settings to control various options, including country and language selection, the number of candidates, and maximum distance of candidates from the point of origin. These options can be controlled through the Spectrum Management Console, Enterprise Designer or through equivalent API methods. See **REST API Features** on page 25 for the information on the API methods.

> **Note:** The GeoComplete component is delivered as a module and can be built into Spectrum stages just like any other component.

GeoComplete Options

The following GeoComplete options can be controlled from Management Console and Enterprise Designer.

> **Note:** The default in Enterprise Designer is to use system defaults. To change an option check the box labeled **Override system default options with the following values**.

- **GeoComplete Address**

  GeoComplete based on a street address.

- **GeoComplete Category**

  GeoComplete based on a point of interest (POI) category or subcategory search.

  > **Note:** No candidates are returned unless you have licensed and installed POI data.

- **GeoComplete POI**

  GeoComplete based on a point of interest.

- **GeoComplete All**

  GeoComplete based on any address or POI criteria.

- **FindNearest POI**

  Find the points of interest nearest to a given location. This is the only search type that cannot use Fuzzy Match .

  or a bounding box search area.
- **Coordinate System**

  EPSG 4326 is currently the only supported coordinate system.

- **Max Candidates**

  Select 1 - 100 to specify the maximum number of candidates returned. If Max Candidates is not specified the default is 5. If you search multiple countries, GeoComplete will consider candidates from each country, sort them internally, and then present the maximum number of candidates based on the combined countries. So whether the default country is set to USA, CAN or CAN,USA, GeoComplete returns the same list of candidates (tied candidates would use country order as a tiebreaker).

  Searching multiple countries using GeoComplete will impact performance.

  Distance Unit

  Specify the distance units in kilometers, miles, meters, or feet. This is associated with the Distance from Origin in the Search Preferences. The default is kilometers.

- **Locale**

Locale is enabled only for FindNearest POI. English is always available and that is the default locale. You can also select Native if the POI categories and subcategories have been localized. Native refers to the local language for that country.

> **Note:** This option is available only if you have licensed and installed POI data.

GeoComplete Search Options

The following GeoComplete Search options can be controlled from Management Console and Enterprise Designer.

- **No Restrictions**

  This selection allows unlimited distance or no bounding box restrictions with GeoComplete searches.

- **Distance from Origin**

  Select this to specify the search extent as the maximum distance from the X/Y origin. This search distance is unlimited by default. The search Distance Unit (kilometers, miles, meters, or feet) is specified on the **GeoComplete** Options tab.

  The distance from origin and the distance units (kilometers or miles) can also be controlled through the API. See **REST API Features** on page 25.

- **Within Bounding Box**

  Select this to specify the X/Y coordinates of a polygon that defines the area in which to search for candidates. The polygon is delimited by four sets of X/Y coordinates defining the corners of the polygon. The corners are defined as:

  - **Upper right longitude**

    The coordinate in the upper right corner of the bounding box that represents the longitude. For example, `-97.149410`.

  - **Lower left longitude**

    The coordinate in the lower left corner of the bounding box that represents the longitude. For example, `-97.615642`.

  - **Upper right latitude**

    The coordinate in the upper right corner of the bounding box that represents the latitude. For example, `30.351415`.

  - **Lower left latitude**

    The coordinate in the lower left corner of the bounding box that represents the latitude. For example, `30.115302`.

From these coordinates, GeoComplete constructs a rectangle that defines the boundaries that will be searched. For example, the coordinates specified above constructs a bounded area centered around Elgin, Texas.

This extent of the search polygon can also be controlled through the API. See **REST API Features** on page 25.

- **Fuzzy Match Settings**

Fuzzy Match searching provides ways to compare input words with intended words and retrieve candidates based on that analysis. You can specify **Hard_Match** or **Soft_Match** for Fuzzy Match, or **None** if you do not want to use Fuzzy Match. This selection is not available if you use the FindNearest POI Search Type. See **Fuzzy Matching Features** on page 13.

- **Match on address number**

You can specify **Match on address number** to determine if a house number match is required to get a match. If this match restriction is checked, then returned candidates must match the input house number. By default, the **Match on address number** box is unchecked, which means that returned candidates do not have to match the input house number.

If the input does not contain a house number, the **Match on address number** restriction has no effect.

GeoComplete Data Preferences

The following GeoComplete Database resource options can be controlled from Management Console and Enterprise Designer.

- **Database**

Provide the name of the database.

- **Dictionary preference**

Select a single dictionary to use or a preferred dictionary. The default is Use Both Dictionaries, which means that both standard and user dictionaries are used with the best close match returned from either dictionary. The choices are:

  - Both standard and user dictionary (Default)
  - Only user dictionary
  - Only standard dictionary
  - Prefer user dictionary
  - Prefer standard dictionary

- **Order Installed Data Resources**

If you have multiple database resources installed, you can select the order in which they are searched. For example, if you had both the USA and CAN databases installed, you could order CAN first and USA second. This applies to the API and Enterprise Designer only.

Returned Candidate Fields

Additional Returned Fields

Returned candidates can be previewed in the Management Console. Candidates include the complete address elements that you expect to see in Spectrum™ Technology Platform candidates, such as AddressLine, City, County, State, and Country. It can also return matched fields, indicating the input fields that were matched in the returned suggested candidates.

GeoComplete Additional Returned Fields

Field Name	Description
AddressLine1	Complete first line of address, typically including street and house number.
AddressNumberMatch	Returns true to indicate that the input address number matched the candidate. If address number is not matched, this field is not returned.
AreaName1Match	Returns true to indicate that the input cstate or province matched the candidate. If it is not matched, this field is not returned.
AreaName2Match	Returns true to indicate that the input county, region or district matched the candidate. If it is not matched, this field is not returned.
AreaName3Match	Returns true to indicate that the input city or town matched the candidate. If it is not matched, this field is not returned.
AreaName4Match	Returns true to indicate that the input locality matched the candidate. If it is not matched, this field is not returned.
City	City or town name.
County	County, region or district
Country	Country 3-letter ISO code.

Field Name	Description
Distance	Distance from origin in kilometers or miles. A distance of -0.0 means that an origin was not used.
DistanceUnit	Distance unit (kilometers or miles)
featureID	unique FeatureID
FormattedString	Formatted address string
isUserDictionary	True if candidate comes from a user dictionary, false if not from user dictionary.
Latitude	Latitude coordinate of candidate. For POI candidates, these are exact coordinates. For street candidates, these coordinates are approximate. To get a more exact coordinates, you must perform a geocode operation on the street candidate.
Longitude	Longitude coordinates of candidate. Coordinates are exact or approximate, as described for Latitude field.
Locality	Name of the locality, suburb or village
miCode	For POI candidates, this miCode is an eight-digit numeric code that identifies POI features by category, subcategory, and subfeature. This is not the same as the featureID and is not related to the Category and Subcategory fields described above. See **miCodes** on page 33 for a comprehensive listing of miCodes that are used for all countries.
Name	Name of the point of interest.
PlaceNameMatch	Returns true to indicate that the input place name matched the candidate. If place name is not matched, this field is not returned.
PostalCode	Postal code
StateProvince	State or province name
StreetName	Street name

Field Name	Description
StreetNameMatch	Returns true to indicate that the input street name matched the candidate. If street name is not matched, this field is not returned.
Type	Returns 1 for a POI match. Returns 2 for a street address match.

Typical Application Examples

In a typical mobile application, you would allow the application to identify your geographic location. Your location will also determine the default country. For the purposes of these examples, we will assume that the default is USA, but you can change this to any supported country. See **GeoComplete Coverage** on page 6 for a list of available countries.

See **Using the APIs** on page 25 for a description of all the options than can be controlled through the API.

Note: The GeoComplete Module is packaged with a sample application. See **GeoComplete Module Features** on page 8 for instructions on how to run the sample application. The sample application has limited capabilities and is for demonstration purposes only. It is not a supported application.

Numbered Street Examples

The following example describes a GeoComplete example with a New York City numbered street.

• Start with focus in USA, New York City, midtown Manhattan. Type: `60 east 4 s`

This returns:

```
60 E 4th St, New York, NY 10003

60 E 4th St, Brooklyn, NY 11218
```

The New York (Manhattan) address is returned first, because this is closest to the origin. An identical address exists in Brooklyn, NY, several miles further from the origin. Others addresses are returned in increasingly further distance from the X/Y origin.

The following example describes a GeoComplete example with a numbered street in Italy.

• Start with focus in Italy (ITA) in the city of Palermo. Type: `via 4`

This returns:

```
Via 4 Aprile, 90133 Palermo
```

A total of ten "via 4" and "via Quarto" street addresses are returned in increasing distance from origin. The word Quarto is recognized as the number 4 in street name.

Street and Directional Abbreviation Examples

• Start with focus in Canada (CAN). Type: `990 north ma`

This returns:

```
990 N Mary Lake Rd, Huntsville, ON, P1H
```

Notice that you entered "North" but the address is returned correctly as 990 "N" Mary Lake Rd, as the address appears in the data source.

• Start with focus in USA, in Nashua, NH. Type: `923 St`

This returns:

```
923 Saint James Pl, Nashua, NH 03062
```

In this example, St is recognized as street abbreviation for Saint.

2 - API

In this section

Using the APIs

This section describes the GeoComplete REST and SOAP APIs.

REST API Features

The module includes REST API methods to control and customize defaults and options.

Table 5: REST API

REST Method	Description
Data.AddressLine1	Single line input
Data.Category	Point of interest category
Data.City	City to search. Combine with AddressLine1 to return candidates in your targeted area.
Data.Country	Specifies the country to search (3-character country ISO code).
Data.OriginLatitude	Y Coordinate of origin.
Data.OriginLongitude	X Coordinate of origin.
Data.PostalCode	Postcode
Data.StateProvince	State or province
Option.BBoxEast	The coordinate defining the eastern boundary of the bounding box.
Option.BBoxNorth	The coordinate defining the northern boundary of the bounding box.
Option.BBoxSouth	The coordinate defining the southern boundary of the bounding box.
Option.BBoxWest	The coordinate defining the western boundary of the bounding box.

REST Method	Description
Option.BoundingBox	Limits the search to the coordinates of the bounding box
Option.CoordinateSystem	Coordinate system to use. Currently only EPSG:4326 is supported.
Option.Database	Specifies the name of the database configured in Management Console. This is required if more than one database is available.
Option.DatabaseSearchOrder	Specifies the order in which to search the databases if more than one database is available. Normally, the databases are searched in the loaded order. For example, the order could be entered as 2, 1, 3, 4 to override the default order.
Option.DefaultCountryCode	Default country if Data.Country is not specified (3-character country ISO code).
Option.DistanceFromOrigin	The extent of the search area as the maximum distance from the X/Y origin.
Option.FuzzyMatch	Hard_Match, Soft_Match, or None to specify the type of fuzzy match. The default is None.
Option.GeoCompleteDictionaryPreference	Specifies the dictionary preference for address dictionary or user dictionary (if you have a custom user dictionary for the country). Values are AD_ONLY (address dictionary only), UD_ONLY (user dictionary only), PREFER_AD, (prefer address dictionary), PREFER_UD, (prefer user dictionary), or BOTH_AD_UD (use both with no preference). The default is BOTH_AD_UD.
Option.GeoCompleteLevel	The GeoComplete search level. This can be GeoCompleteAddress, GeoCompletePOI, GeoCompleteCategory, GeoCompleteALL, or FindNearestPOI. The default is GeoCompleteALL. Unless you have licensed and installed POI data, GeoCompletePOI, GeoCompleteCategory, and FindNearestPOI will not return any candidates and only street candidates will be returned for GeoCompleteAll.
Option.Locale	Locale is enabled only for FindNearest POI. English is always available and that is the default locale. You can also select Native if the POI categories and subcategories have been localized. Native refers to the local language for that country. **Note:** This option is available only if you have licensed and installed POI data.
Option.MatchOnAddressNumber	Specifies whether an address number match is required. By default, an address number match is not required.
Option.MaxCandidates	Maximum number of candidates to return (1 - 100). Or specify -1 to return the maximum allowable number of candidates (100).

REST Method	Description
Option.NoRestriction	Allows no distance restrictions with GeoComplete searches.
Option.Path	Option path.
Option.SearchDistance	Specifies the distance (radius) to search for candidates from the given origin. Distance units are defined by the SearchRadiusUnits method. This distance is also used as a search distance restriction.
Option.SearchRadiusUnit	Units used for Option.SearchDistance. Values can be (KILOMETERS, MILES, METERS, or FEET). The default is KILOMETERS, unless you changed that preference in Management Console.

REST API Example

This example illustrates a REST API request and JSON response using multiple fields to return candidates.

See **REST API Features** on page 25 for a description of all the options than can be controlled through the API.

When you choose a candidate, the address can be placed on a map.

Street Address Example with Multiple Field Support

The following example illustrates a request for candidates using multiline support. It begins with a partial address (Data.AddressLine1=Jordan). The response included candidates from all over the US.

```
http://localhost:8080/rest/GeoComplete/results.json?Data.AddressLine1=jordan&Data.Country=USA&Option.Database=GCM
```

By adding Data.PostalCode=12180 to the request the returned candidates are more targeted.

```
http://localhost:8080/rest/GeoComplete/results.json?Data.AddressLine1=jordan&Data.Country=USA&Option.Database=GCM&Data.PostalCode=12180
```

```
{
        "output_port" : [ {
        "AddressLine1" : "Jordan Rd",
        "City" : "Troy",
        "Country" : "USA",
        "LastLine" : "Troy, NY 12180",
        "Latitude" : "42.679065",
        "Longitude" : "-73.692485",
```

```
            "PostalCode" : "12180",
            "StateProvince" : "NY",
            "Type" : "2",
            "Distance" : "-0.0",
            "DistanceUnit" : "KILOMETERS",
            "FormattedString" : "Jordan Rd, Troy, NY 12180",
            "IsUserDictionary" : "false",
            "featureID" : "368400000856999",
            "StreetNameMatch" : "true",
            "user_fields" : [ ]
            }, {
            "AddressLine1" : "Jordan Point Dr",
            "City" : "Troy",
            "Country" : "USA",
            "LastLine" : "Troy, NY 12180",
            "Latitude" : "42.676154999999994",
            "Longitude" : "-73.688185",
            "PostalCode" : "12180",
            "StateProvince" : "NY",
            "Type" : "2",
            "Distance" : "-0.0",
            "DistanceUnit" : "KILOMETERS",
            "FormattedString" : "Jordan Point Dr, Troy, NY 12180",
            "IsUserDictionary" : "false",
            "featureID" : "368400001448931",
            "StreetNameMatch" : "true",
            "user_fields" : [ ]
            } ]
            }
```

Sample SOAP Request and Response

This topic shows sample SOAP requests and responses.

> **Note:** Since the service name, option name, and field name ultimately become XML elements, they may not contain characters that are invalid in XML element names (for example, spaces are not valid). Please consult the xml specification for clarification (**XML Markup Language**). Services not meeting this requirement will still function but will not be exposed as a web services.

Sample SOAP Request

Following is a sample SOAP request and response. The sections for options and rows are all dependent on the metadata for that particular web service; therefore, different components will have

different metadata entries. Additionally, there is a user_fields section that allows you to pass in field values that will be returned, unmodified, in the response.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:geoc="http://www.pb.com/spectrum/services/GeoComplete">
    <soapenv:Header/>
    <soapenv:Body>
        <geoc:GeoCompleteRequest>
            <!--Optional:-->
            <geoc:options>
                <!--Optional:-->
                <geoc:Database>bel</geoc:Database>
                <!--Optional:-->
                <geoc:DatabaseSearchOrder></geoc:DatabaseSearchOrder>
                <!--Optional:-->
                <geoc:Path></geoc:Path>
                <!--Optional:-->
                <geoc:GeoCompleteLevel></geoc:GeoCompleteLevel>
                <!--Optional:-->
                <geoc:NoRestriction></geoc:NoRestriction>
                <!--Optional:-->
                <geoc:DistanceFromOrigin></geoc:DistanceFromOrigin>
                <!--Optional:-->
                <geoc:SearchDistance></geoc:SearchDistance>
                <!--Optional:-->
                <geoc:BoundingBox></geoc:BoundingBox>
                <!--Optional:-->
                <geoc:BBoxEast></geoc:BBoxEast>
                <!--Optional:-->
                <geoc:BBoxWest></geoc:BBoxWest>
                <!--Optional:-->
                <geoc:BBoxNorth></geoc:BBoxNorth>
                <!--Optional:-->
                <geoc:BBoxSouth></geoc:BBoxSouth>
                <!--Optional:-->
                <geoc:MatchOnAddressNumber></geoc:MatchOnAddressNumber>
                <!--Optional:-->

<geoc:GeoCompleteDictionaryPreference></geoc:GeoCompleteDictionaryPreference>

                <!--Optional:-->
                <geoc:CoordinateSystem></geoc:CoordinateSystem>
                <!--Optional:-->
                <geoc:MaxCandidates></geoc:MaxCandidates>
                <!--Optional:-->
                <geoc:SearchRadiusUnit></geoc:SearchRadiusUnit>
                <!--Optional:-->
                <geoc:Locale></geoc:Locale>
                <!--Optional:-->
                <geoc:FuzzyMatch></geoc:FuzzyMatch>
            </geoc:options>
            <!--Optional:-->
```

```
            <geoc:input_port>
                <!--Zero or more repetitions:-->
                <geoc:Address>
                    <!--You may enter the following 9 items in any order-->
                    <!--Optional:-->
                    <geoc:AddressLine1>Rijksweg 60</geoc:AddressLine1>
                    <!--Optional:-->
                    <geoc:City>Bornem</geoc:City>
                    <!--Optional:-->
                    <geoc:StateProvince>Vlaams Gewest</geoc:StateProvince>
                    <!--Optional:-->
                    <geoc:PostalCode>2880</geoc:PostalCode>
                    <!--Optional:-->
                    <geoc:Country>BEL</geoc:Country>
                    <!--Optional:-->
                    <geoc:Category>Shopping</geoc:Category>
                    <!--Optional:-->
                    <geoc:OriginLatitude></geoc:OriginLatitude>
                    <!--Optional:-->
                    <geoc:OriginLongitude></geoc:OriginLongitude>
                    <!--Optional:-->
                    <geoc:user_fields>
                        <!--Zero or more repetitions:-->
                        <geoc:user_field>
                            <geoc:name></geoc:name>
                            <geoc:value></geoc:value>
                        </geoc:user_field>
                    </geoc:user_fields>
                </geoc:Address>
            </geoc:input_port>
        </geoc:GeoCompleteRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

Sample SOAP Response

Below is a sample SOAP response.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns3:GeoCompleteResponse xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/GeoComplete">
            <ns3:output_port>
                <ns3:Address>
                    <ns3:AddressLine1>Rijksweg 60</ns3:AddressLine1>
                    <ns3:City>Bornem</ns3:City>
                    <ns3:County>Mechelen</ns3:County>
                    <ns3:Country>BEL</ns3:Country>
                    <ns3:LastLine>2880 Bornem</ns3:LastLine>
                    <ns3:Latitude>51.09881</ns3:Latitude>
                    <ns3:Longitude>4.2529</ns3:Longitude>
                    <ns3:PostalCode>2880</ns3:PostalCode>
```

```
            <ns3:StateProvince>Vlaams Gewest</ns3:StateProvince>
            <ns3:Name>DreamLand</ns3:Name>
            <ns3:Type>1</ns3:Type>
            <ns3:Distance>-0.0</ns3:Distance>
            <ns3:DistanceUnit>KILOMETERS</ns3:DistanceUnit>
            <ns3:FormattedString>DreamLand, Rijksweg 60, 2880
Bornem</ns3:FormattedString>
            <ns3:BrandName>DreamLand</ns3:BrandName>
            <ns3:IsUserDictionary>false</ns3:IsUserDictionary>
            <ns3:Category>Shopping</ns3:Category>
            <ns3:SubCategory>Toys and Games</ns3:SubCategory>
            <ns3:micode>10010374</ns3:micode>
            <ns3:http>www.dreamland.be</ns3:http>
            <ns3:telnum>+(32)-(3)-8254566</ns3:telnum>
            <ns3:email>dreamland@dreamland.be</ns3:email>
            <ns3:featureID>10560201478679</ns3:featureID>
            <ns3:StreetNameMatch>true</ns3:StreetNameMatch>
            <ns3:AddressNumberMatch>true</ns3:AddressNumberMatch>
            <ns3:AreaName1Match>true</ns3:AreaName1Match>
            <ns3:AreaName3Match>true</ns3:AreaName3Match>
            <ns3:CategoryMatch>true</ns3:CategoryMatch>
            <ns3:user_fields>
                <ns3:user_field>
                    <ns3:name/>
                    <ns3:value/>
                </ns3:user_field>
            </ns3:user_fields>
         </ns3:Address>
       </ns3:output_port>
     </ns3:GeoCompleteResponse>
   </soap:Body>
</soap:Envelope>
```

A - Point of Interest Codes (miCodes)

In this section

miCodes

Point of Interest candidates return the miCode. These miCodes are eight-digit numeric codes that provide feature identification by general category, subcategory, and subfeature. Point of Interest miCodes are organized with the following numeric pattern:

> **Note:** POI and category/subcategory features are available only if you have licensed and installed POI data.

POI miCode Numeric Patterns

10 - all Point of Interest miCodes begin with the digits 10

Digits 3 and 4 identify the POI subcategory (such as Shopping - Retail and Wholesale; Health

Digits 5 through 8 identify the subfeature codes (such as Food Stores; Hospital/Polyclinic)

Point of Interest miCodes are assigned universally across all countries.

> **Note:** POI data is a separate data bundle that must be purchased.

Table 6: POI miCodes

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
00 Unclassified	00 Unclassified	10000000
01 Shopping - Retail and Wholesale	0000 Unclassified	10010000
	0000 Unclassified	10010000
	0100 General Merchandise Stores/Shops	10010100
	0101 Department Stores	10010101
	0102 Variety Stores	10010102
	0103 Warehouse Club Stores	10010103
	0200 Food Stores	10010200

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0201 Food Stores - Supermarket	10010201
	0203 Food Stores -Fish/Seafood Market	10010203
	0204 Food Stores -Fruits/Vegetable Market	10010204
	0205 Beverage / Bottle Store	10010205
	0300 Apparel and Accessory	10010300
	0400 Auto Sales and Service Stations	10010400
	0401 Car Dealership - Sales	10010401
	0402 Car Dealership - Service	10010402
	0403 Car Dealership - Sales and Service	10010403
	0404 Car Repair Facilities	10010404
	0405 Car Rental Office	10010405
	0410 Motorcycle Sales and Service	10010410
	0500 Gas / Petrol Stations	10010500
	0600 Furniture and Home Furnishings	10010600
	0700 Building Materials AND Garden Supplies	10010700
	0800 Wholesale Trade Durable AND Non-durables	10010800
	0900 Shopping Malls/ Shopping Centers	10010900

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
02 Eating and Drinking Places	0000 Unclassified	10020000
	0101 Restaurants - Ethnic - American	10020101
	0102 Restaurants - Ethnic - Chinese	10020102
	0103 Restaurants - Ethnic - Italian	10020103
	0104 Restaurants - Ethnic - German	10020104
	1050 Restaurants - Ethnic - French	10020105
	0106 Restaurants - Ethnic - Japanese	10020106
	0107 Restaurants - Ethnic - Indian	10020107
	0108 Restaurants - Ethnic - Thai	10020108
	0109 Restaurants - Ethnic - Vietnamese	10020109
	0110 Restaurants - Ethnic - Middle East	10020110
	0111 Restaurants - Ethnic - Greek	10020111
	0112 Restaurants - Ethnic - Mexican	10020112
	0200 Restaurants - Ice Cream/Soda	10020200
	0300 Restaurants - Fast Food	10020300
	0400 Restaurants - Family	10020400
	0500 Restaurants - Pizza	10020500
	0600 Restaurants - Seafood	10020600
	0700 Restaurants - Drinking/Alcoholic	10020700

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0800 High Concentration of Restaurants	10020800
03 Business and Personal Services	0000 Unclassified	10030000
	0100 Hotels/Motels	10030100
	0200 Personal Service Business	10030200
	0201 Laundromat	10030201
	0202 Salons	10030202
	0300 Financial Services	10030300
	0301 Banks	10030301
	0302 ATM Machine / Cash Dispenser	10030302
	0400 Broadcasting Service	10030400
	0500 Automobile Club	10030500
04 Other Business	0000 Office / Company / Industry	10040000
	0100 Agriculture and Mining	10040100
	0101 Mine/Quarry	10040101
	0200 Manufacturing	10040200
11 Tourism	0000 Unclassified	10110000
	0100 Tourist Attraction	10110100
	0200 Tourist Attraction - Building	10110200
	0201 Museum	10110201

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0202 Art Gallery / Art Center	10110202
	0203 Aquarium	10110203
	0300 Tourist Attraction - Monument	10110300
	0301 Memorial	10110301
	0400 Tourist Attraction - Natural	10110400
	0401 Scenic/Panoramic View	10110401
	0402 Beach, Represented by a Point	10110402
	0403 Mountain peak	10110403
	0405 Mountain pass	10110405
	0500 Tourist Attraction - Historic Site	10110500
	0600 Tourist Attraction - Other	10110600
	0601 Winery	10110601
	0700 Tourist Information Center	10110700
	0800 Rest Area (Public)	10110800
12 Sports	0000 Sports - Unclassified	10120000
	0100 Sport Center / Stadium	10120100
	0101 Tennis Court	10120101
	0102 Cricket	10120102
	0103 Football/Soccer	10120103

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0104 Baseball	10120104
	0105 Ice Skating Rink	10120105
	0106 Hockey	10120106
	0107 Basketball/Netball	10120107
	0108 Golf Course	10120108
	0109 Race Course	10120109
	0110 Rifle Range	10120110
	0111 Bowling Center	10120111
	0300 Water Sport	10120300
	0301 Swimming Pool	10120301
	0302 Yacht Basin	10120302
	0400 Sports Club	10120400
	0500 Health Club	10120500
	0600 Ski Resort	10120600
13 Recreation/Entertainment	0000 Unclassified	10130000
	0100 Cultural /Civic/ Community Center	10130100
	0101 Theater	10130101
	0102 Opera	10130102
	0103 Concert Hall	10130103

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0104 Music Center	10130104
	0105 Cinema	10130105
	0200 Leisure Center	10130200
	0201 Park	10130201
	0202 Zoo	10130202
	0203 Amusement Park	10130203
	0204 Camping Ground/Caravan park	10130204
	0205 Fairgrounds	10130205
	0206 Night Life /Clubs	10130206
	0207 Casino	10130207
	0208 Recreational Clubs	10130208
	0300 Convention Center	10130300
	0301 Exhibition Center	10130301
21 Educational	0000 Educational	10210000
	0100 Kindergarten/Nursery School	10210100
	0200 Elementary/Middle/High School	10210200
	0201 Elementary/Middle/High School - Catholic	10210201
	0202 Elementary/Middle/High School - Private	10210202

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0203 Elementary/Middle/High School - Public	10210203
	0300 Colleges/Universities	10210300
	0400 Vocational/Trade	10210400
	0500 Special Schools	10210500
	0600 Library	10210600
	0700 Child Care Center	10210700
22 Places of Worship	0100 Church	10220100
	0101 Baptist Church	10220101
	0102 Catholic Church	10220102
	0103 Methodist Church	10220103
	0200 Mosque	10220200
	0300 Synagogue	10220300
	0400 Buddhist Temple	10220400
	0500 Hindu Temple	10220500
	0600 Other	10220600
23 Health	0000 Health Facilities, Undefined	10230000
	0010 Pharmacy	10230010
	0100 Hospital/Polyclinic	10230100
	0200 Nursing and Personal Care	10230200

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0300 Office/Clinic of Medical Doctors	10230300
	0400 Office/Clinic of Dentist	10230400
	0500 Medical/Dental Laboratories/Services	10230500
	0600 Ambulance Station	10230600
	0700 Veterinarian Clinic/Hospital	10230700
24 Public	0000 All	10240000
	0100 Government Facilities	10240100
	0101 City/Municipal/Local Facility	10240101
	0102 Courts	10240102
	0103 Embassy /Foreign Government Representation	10240103
	0104 Frontier Crossing	10240104
	0300 Police Station/Facility	10240300
	0400 Fire Stations	10240400
	0500 Post Office	10240500
	0501 Post Office - Main	10240501
	0502 Post Office - Minor	10240502
	0600 Prisons	10240600
	0700 Military base /Defense	10240700

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0800 Cemetery/Crematorium	10240800
	0900 Waste Disposal /Sewerage	10240900
31 Airports	0000 Unclassified	10310000
	0010 Check-in Facility at Airport	10310010
	0011 Check-in Facility at Airport Departures	10310011
	0012 Check-in Facility at Airport Arrivals	10310012
	0010 Check-in Facility at Airport Departure/Arrivals	10310013
	0100 Airport - Civil	10310100
	0101 Airport - Public	10310101
	0102 Airport- Private	10310102
	0200 Airport - Military	10310200
	0300 Helipad	10310300
32 Stations	0000 Unclassified	10320000
	0100 Ferry terminal /Wharf/Jetty/Pontoon	10320100
	0101 Ferry Terminal Ship/Hovercraft	10320101
	0102 Ferry Terminal Train	10320102
	0200 Railway Station, Undefined	10320200
	0201 Railway Station, Underground/Metro	10320201

10 Points Of Interest Subcategories	Subfeature Codes	miCodes
	0202 Railway Station, International	10320202
	0203 Railway Station, National	10320203
	0204 Railway Station, (Sub) Urban	10320204
	0300 Bus Terminal /Depot	10320300
33 Parking	0000 Parking - Unclassified	10330000
	0100 Parking Garage	10330100
	0200 Open Parking Area	10330200
	0300 Rent-a-Car Parking	10330300
	0210 Park and Ride Lot	10330210
34 Interchanges	0000 Intersection	10340000
	0100 Freeway Exit/Entrance	10340100
	0200 Freeway to Freeway Interchange	10340200
	0300 Non Freeway Intersection	10340300

Notices

USPS® Notices

Pitney Bowes Inc. holds a non-exclusive license to publish and sell ZIP + 4® databases on optical and magnetic media. The following trademarks are owned by the United States Postal Service: CASS, CASS Certified, DPV, eLOT, FASTforward, First-Class Mail, Intelligent Mail, LACS^Link, NCOA^Link, PAVE, PLANET Code, Postal Service, POSTNET, Post Office, RDI, Suite^Link , United States Postal Service, Standard Mail, United States Post Office, USPS, ZIP Code, and ZIP + 4. This list is not exhaustive of the trademarks belonging to the Postal Service.

Pitney Bowes Inc. is a non-exclusive licensee of USPS® for NCOA^Link® processing.

Prices for Pitney Bowes Software's products, options, and services are not established, controlled, or approved by USPS® or United States Government. When utilizing RDI™ data to determine parcel-shipping costs, the business decision on which parcel delivery company to use is not made by the USPS® or United States Government.

Data Provider and Related Notices

Data Products contained on this media and used within Pitney Bowes Software applications are protected by various trademarks and by one or more of the following copyrights:

© Copyright United States Postal Service. All rights reserved.

© 2014 TomTom. All rights reserved. TomTom and the TomTom logo are registered trademarks of TomTom N.V.

© 2016 HERE

Fuente: INEGI (Instituto Nacional de Estadística y Geografía)

Based upon electronic data © National Land Survey Sweden.

© Copyright United States Census Bureau

© Copyright Nova Marketing Group, Inc.

Portions of this program are © Copyright 1993-2007 by Nova Marketing Group Inc. All Rights Reserved

© Copyright Second Decimal, LLC

© Copyright Canada Post Corporation

This CD-ROM contains data from a compilation in which Canada Post Corporation is the copyright owner.

© 2007 Claritas, Inc.

The Geocode Address World data set contains data licensed from the GeoNames Project (**www.geonames.org**) provided under the Creative Commons Attribution License ("Attribution

License") located at **http://creativecommons.org/licenses/by/3.0/legalcode**. Your use of the GeoNames data (described in the Spectrum™ Technology Platform User Manual) is governed by the terms of the Attribution License, and any conflict between your agreement with Pitney Bowes Software, Inc. and the Attribution License will be resolved in favor of the Attribution License solely as it relates to your use of the GeoNames data.