Spectrum Technology Platform

Version 9.0 SP3

REST Web Services Guide

Pitney Bowes
Software

Contents

1

Getting Started

In this section:

The REST Interface

Spectrum™ Technology Platform provides a REST interface to web services. User-defined web services, which are those created in Enterprise Designer, support GET and POST methods. Default services installed as part of a module only support GET. If you want to access one of these services using POST you must create a user-defined service in Enterprise Designer.

To view the REST web services available on your Spectrum™ Technology Platform server, go to:

`http://server:port/rest`

Note the following limitations on REST web services:

- Web services support GET and POST methods. The methods available for a specific web service are configured in the service's web service options in Enterprise Designer.
- We recommend that you limit parameters to 2,048 characters due to URL length limits.

Service Endpoints

The default endpoints for REST services are:

`http://server:port/rest/service_name/results.xml`

`http://server:port/rest/service_name/results.json`

Endpoints for user-defined web services can be modified in Enterprise Designer to use a different URL.

Note: By default Spectrum™ Technology Platform uses port 8080 for HTTP communication. Your administrator may have configured a different port.

WADL URL

The WADL for Spectrum™ Technology Platform web services is:

`http://server:port/rest/service_name?_wadl`

For example:

`http://myserver:8080/rest/ValidateAddress?_wadl`

User Fields

You can pass extra fields through the web service even if the web service does not use them. These fields are returned, unmodified, in the `user_fields` section of the response. For GET requests, user fields are passed in as a parameter in the URL like any other field. For POST requests, user fields are passed in as part of the `user_fields` element in the XML or JSON request.

Note: User field names may not contain characters that are invalid in XML or JSON element names. For example, spaces are not valid.

Sample REST Request Using GET with XML Response

The following example illustrates how to make a REST request to the ValidateAddress service using the GET method requesting a response in XML.

```
http://localhost:8080/rest/ValidateAddress/results.xml?Option.OutputCasing=U&
Data.AddressLine1=1825+Kramer+Lane&Data.PostalCode=78759
```

The sample request would result in this response since an XML response was requested:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xml.ValidateAddressResponse
xmlns="http://www.pb.com/spectrum/services/ValidateAddress">
    <output_port>
        <Address>
            <Confidence>82</Confidence>
            <RecordType>Normal</RecordType>
            <CountryLevel>A</CountryLevel>
            <ProcessedBy>USA</ProcessedBy>
            <MatchScore>0</MatchScore>
            <AddressLine1>1825 KRAMER LN</AddressLine1>
            <City>AUSTIN</City>
            <StateProvince>TX</StateProvince>
            <PostalCode>78758-4260</PostalCode>
            <PostalCode.Base>78758</PostalCode.Base>
            <PostalCode.AddOn>4260</PostalCode.AddOn>
            <Country>UNITED STATES OF AMERICA</Country>
            <user_fields/>
        </Address>
    </output_port>
</xml.ValidateAddressResponse>
```

Sample REST Request Using GET with JSON Response

The following example illustrates how to make a REST request to the ValidateAddress service using the GET method requesting a response in JSON.

```
http://localhost:8080/rest/ValidateAddress/results.json?Option.OutputCasing=U&
Data.AddressLine1=1825+Kramer+Lane&Data.PostalCode=78759
```

The sample request wold result in this response since a JSON response was requested:

```
{
 "ns1.json.ValidateAddressResponse" :
 {
 "ns1.output_port" :
 {
 "ns1.Confidence" : 82,
 "ns1.RecordType" : "Normal",
 "ns1.CountryLevel" : "A",
 "ns1.ProcessedBy" : "USA",
 "ns1.MatchScore" : 0,
 "ns1.AddressLine1" : "1825 KRAMER LN",
 "ns1.City" : "AUSTIN",
 "ns1.StateProvince" : "TX",
 "ns1.PostalCode" : "78758-4260",
 "ns1.PostalCode.Base" : 78758,
 "ns1.PostalCode.AddOn" : 4260,
 "ns1.Country" : "UNITED STATES OF AMERICA"
 }
 }
}
```

Related Links

Exposing a Service as a Web Service on page 11

JSON POST Request

User-defined web services can be exposed as a REST web service and configured to have a POST method that accepts JSON input. Use the following format for JSON POST requests.

Flat Data

Use this format to send flat data to a web service using POST:

```
{
  "InputStageName":
  {
    "InputDataType": [
      {
        "FieldName1": "FieldValue1",
        "FieldName2": "FieldValue2"
      }
    ]
  }
}
```

Where:

InputStageName

The name of the input stage as shown on the canvas in Enterprise Designer. The default name of the stage is `Input`.

InputDataType

The name given to the record-level entity. This value is specified in the dataflow's Input stage, in the **Data type name** field on the **Input Fields** tab. The default name of the record-level entity is `Row`.

FieldName1* and *FieldName2

The names of the input fields defined in the service's Input stage.

FieldValue1* and *FieldValue2

Input data that you want to send to the web service in the corresponding field.

List Data

List data consists of hierarchical groupings of fields grouped under a parent field.

Note: In order to use list data as input, the service must be exposed as a REST web service without any GET resources. If the service has a GET resource you will get an error in Enterprise Designer when exposing the service because hierarchical fields are not supported for GET.

Use the following format to send list data to a web service using POST.

```
{
  "InputStageName":
  {
    "InputDataType": [
      {
        "ListField1":[
          {"SubfieldName1": "SubfieldValue1"},
          {"SubfieldName2": "SubfieldValue2"}
        ]
      }
    ]
  }
}
```

Where:

InputStageName

The name of the input stage as shown on the canvas in Enterprise Designer. The default name of the stage is `Input`.

InputDataType

The name given to the record-level entity. This value is specified in the dataflow's Input stage, in the **Data type name** field on the **Input Fields** tab. The default name of the record-level entity is `Row`.

ListField1

The name of the hierarchical field defined in the service's Input stage.

SubfieldName1* and *SubfieldName2

The names of child fields that comprise the list field.

SubfieldValue1* and *SubfieldValue2

Input data that you want to send to the web service.

User Fields

You can pass extra fields through the web service even if the web service does not use them. These fields are returned, unmodified, in the `user_fields` section of the response. The user fields you supply in the request do not need to be defined in the service dataflow's Input stage.

```
{
  "InputStageName":
  {
    "InputDataType": [
      {
        "user_fields": [
        {
          "name": "FieldName1",
          "value": "FieldValue1"
        },
        {
          "name": "FieldName2",
          "value": "FieldValue2"
        }]
      }
    ]
  }
}
```

Where:

InputStageName

The name of the input stage as shown on the canvas in Enterprise Designer. The default name of the stage is `Input`.

InputDataType

The name given to the record-level entity. This value is specified in the dataflow's Input stage, in the **Data type name** field on the **Input Fields** tab. The default name of the record-level entity is `Row`.

FieldName1* and *FieldName2

The name of the pass-through field.

FieldValue1* and *FieldValue2

The data you want to include in the passthrough field.

Options

You can specify options in the request, overriding the default options specified in the service dataflow. For user-defined web services, you can only specify options in the request if the dataflow has been configured to accept options. To configure a service to accept options in the request, open the service in Enterprise Designer and select **Edit** > **Dataflow Options**.

To specify processing options in a request, use this format:

```
"options" : {
     "OptionName1" : "Value1"
},
```

Where:

OptionName1

The name of the option. For a list of valid options for the service see the service's WADL or open the service in Enterprise Designer and select **Edit** > **Dataflow Options**.

OptionValue1

A legal value for the option. For a list of legal values, open the service in Enterprise Designer and select **Edit** > **Dataflow Options**.

Example JSON Request using POST

The following example demonstrates how to include options, flat fields, a list field, and user-defined fields in a JSON request to a web service using POST.

```
{
  "options" : {
    "OutputCasing" : "U"
},
  "Input":
  {
    "Address": [
      {
        "AddressLine1": "1825 Kramer Ln",
        "City": "Austin",
        "StateProvince": "TX",
        "Accounts": [
        {

          "AccountNumber": "120993",
          "ExpirationDate": "10-3-2017"
         },
         {

          "AccountNumber": "898732",
          "ExpirationDate": "8-13-2016"
         }
        ],
        "user_fields": [
        {
          "name": "Note1",
          "value": "Prefers decaffeinated coffee"
        },
        {
          "name": "Note2",
          "value": "Requests east facing window"
        }]
      }
    ]
  }
}
```

In this example,

- `OutputCasing` is an option exposed by the web service that controls whether the output is returned in upper case or lower case. In this request, it is set to U for upper case.
- `Input` is the label of the Input stage in the dataflow as displayed on the canvas in Enterprise Designer.

- `Address` is the name of the record-level entity as specified in the dataflow's Input stage, in the **Data type name** field on the **Input Fields** tab.
- `AddressLine1`, `City`, and `StateProvince` are flat fields.
- `Accounts` is a hierarchical ("list") field containing subfields name `AccountNumber` and `ExpirationDate`. There are two accounts included in this example.
- `user_fields` contains user-defined fields that are passed through and returne in the output unmodified by the web service.

Exposing a Service as a Web Service

Spectrum™ Technology Platform services can be made available as REST and/or SOAP web services. To make a service available on your server as a web service:

1. Open Enterprise Designer.
2. Open the service that you want to expose as a web service.
3. Go to **Edit** > **Web Service Options**.
4. To make the service available as a SOAP web service, check the box **Expose as SOAP web service**.
5. To make the service available as a REST web service, check the box **Expose as REST web service** and complete the following steps.
   a) If you want to override the default endpoint, specify the endpoint you want to use in the **Path** field.

      Specifying a path is optional. By default, a REST web service's endpoint is:

      `http://`*`server`*`:`*`port`*`/rest/`*`service_name`*`/results.`*`qualifier`*

      If you want to use a different endpoint, the path you specify is added after the service name. For example, if you specify `Americas/Shipping` in the **Path** field, your JSON endpoint would be something like this:

      `http://myserver:8080/rest/MyService/`**`Americas/Shipping`**`/results.json`

      You can use fields and options from the dataflow as variable names in the path by clicking the **Insert variable** drop-down menu and selecting the field or option you want to use. The variable is represented in the path using the notation `${Option.`*`Name`*`}` for dataflow options and `${Data.`*`Name`*`}` for dataflow fields.

   b) By default REST web services support the GET method and return data in XML and JSON formats. You can define additional HTTP methods and output formats by clicking **Add** to add a resource to the web service.

      When you add a resource, you can choose the HTTP method (**GET** or **POST**). The supported data formats are listed below. You may not have all these formats available to you because some formats are only available if you have certain modules installed on your Spectrum™ Technology Platform server.

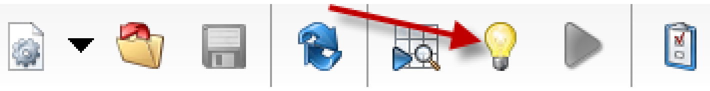
XML	The default XML format. Use this format if you want to use XML as the format for requests and responses, and there is no specialized XML format for the kind of data you want to process.
JSON	The default JSON format. Use this format if you want to use JSON as the format for requests and responses, and there is no specialized JSON format for the kind of data you want to process.

   c) Click **OK**.

      The new resource is added to the web service.

6. Click **OK** when you are done configuring the web service options.
7. Click the gray light bulb in the tool bar to expose the service.

When a dataflow is exposed the light bulb button in the Enterprise Designer tool bar indicates that the dataflow is exposed as shown here:



To verify that the service is now exposed as a web service, go to one of the following URLs:

- For REST: `http://`*`server`*`:`*`port`*`/rest`
- For SOAP: `http://`*`server`*`:`*`port`*`/soap`

Where *server* is the name or IP address of your Spectrum™ Technology Platform server and *port* is the port used for HTTP communication.

Related Links

The REST Interface on page 6

Sample Web Application

The following sample class for .NET is written in C# on Visual Studio 2010. Proxy class implementations for the web service data types ValidateAddressClient, requestRow, context , options and responseRow were generated using Visual Studio .NET's "Add Service Reference" command. This example uses the ValidateAddress web service. It is important to note that in this example, the appropriate credentials must be provided or the call will fail.

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Net;
using System.Text;
using ConsoleApplication1.ValidateAddress_Reference;

namespace Test
{
    class Program
    {
        static void Main(string[] args)
        {
            var validateClient = new ValidateAddress {Credentials = new
NetworkCredential("admin", "admin")};

            var address1 = new input_portAddress
            {
                AddressLine1 = "1825B Kramer Lane",
                AddressLine2 = "Suite 100",
                PostalCode = "78758",
                City = "Austin",
                StateProvince = "Texas"
            };

            var address2 = new input_portAddress
            {
                AddressLine1 = "100 Congress",
                PostalCode = "78701",
                City = "Austin",
                StateProvince = "Texas"
            };

            var addresses = new input_portAddress[2];
            addresses[0] = address1;
            addresses[1] = address2;
            var options = new options {OutputCasing = OutputCasing.M};
            output_portAddress[] results =
validateClient.CallValidateAddress(options, addresses);

            for (int i = 0; i < results.Length; i++)
            {
                System.Console.WriteLine("Record " + (i+1) + ":");
                System.Console.WriteLine("AddressLine1=" +
results[i].AddressLine1);
                System.Console.WriteLine("City=" + results[i].City);
                System.Console.WriteLine("StateProvince=" +
results[i].StateProvince);
                System.Console.WriteLine("PostalCode=" +
results[i].PostalCode + "\n");
            }

            System.Console.Write("Press any key to continue...");
            System.Console.ReadKey();
        }
    }
}
```

2

Web Services

In this section:

Address Now Module

BuildGlobalAddress

BuildGlobalAddress allows you to build a valid address starting with just a single address element or a few address elements. BuildGlobalAddress is part of the Address Now Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/BuildGlobalAddress/results.json
```

XML endpoint:

```
http://server:port/rest/BuildGlobalAddress/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/BuildGlobalAddress/results.json?
Data.Action=init&Data.Country=USA
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Action": "init",
   "Country": "USA",
   "SessionId": "n5hiliawx0",
   "SearchFieldIndex": "-1",
   "Field.0.Name": "Zip",
   "Field.0.Index": "0",
   "Field.0.Value": "",
   "Field.0.CommitFlag": "N",
   "Field.1.Name": "City",
   "Field.1.Index": "1",
   "Field.1.Value": "",
   "Field.1.CommitFlag": "N",
   "Field.2.Name": "State",
   "Field.2.Index": "2",
   "Field.2.Value": "",
   "Field.2.CommitFlag": "N",
   "Field.3.Name": "County",
   "Field.3.Index": "3",
   "Field.3.Value": "",
   "Field.3.CommitFlag": "N",
   "Field.4.Name": "Street",
   "Field.4.Index": "4",
   "Field.4.Value": "",
   "Field.4.CommitFlag": "N",
   "Field.5.Name": "Company",
   "Field.5.Index": "5",
   "Field.5.Value": "",
   "Field.5.CommitFlag": "N",
   "Field.6.Name": "Building",
   "Field.6.Index": "6",
   "Field.6.Value": "",
   "Field.6.CommitFlag": "N",
   "Field.7.Name": "Premise-No",
   "Field.7.Index": "7",
   "Field.7.Value": "",
   "Field.7.CommitFlag": "N",
   "Field.8.Name": "Sub-Building",
   "Field.8.Index": "8",
   "Field.8.Value": "",
   "Field.8.CommitFlag": "N",
   "Field.9.Name": "PO-Box",
   "Field.9.Index": "9",
   "Field.9.Value": "",
   "Field.9.CommitFlag": "N",
   "Field.10.Name": "+4",
   "Field.10.Index": "10",
   "Field.10.Value": "",
   "Field.10.CommitFlag": "N",
   "Alternatives.InContext": "",
   "Alternatives.InContext.Count": "0",
   "Alternatives.OutContext": "",
   "Alternatives.OutContext.Count": "0",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response. This request performs the initialization (`init`) action to obtain a session ID from BuildGlobalAddress.

```
http://myserver:8080/rest/BuildGlobalAddress/results.xml?
Data.Action=init&Data.Country=USA
```

The XML returned by this request would be:

```
<ns2:xml.BuildGlobalAddressResponse
xmlns:ns2="http://www.pb.com/spectrum/services/BuildGlobalAddress">
   <ns2:output_port>
      <ns2:Result>
         <ns2:Action>init</ns2:Action>
         <ns2:Country>USA</ns2:Country>
         <ns2:SessionId>qbximydxf0</ns2:SessionId>
         <ns2:Field.0.Name>Zip</ns2:Field.0.Name>
         <ns2:Field.0.Index>0</ns2:Field.0.Index>
         <ns2:Field.0.Value/>
         <ns2:Field.0.CommitFlag>N</ns2:Field.0.CommitFlag>
         <ns2:Field.1.Name>City</ns2:Field.1.Name>
         <ns2:Field.1.Index>1</ns2:Field.1.Index>
         <ns2:Field.1.Value/>
         <ns2:Field.1.CommitFlag>N</ns2:Field.1.CommitFlag>
         <ns2:Field.2.Name>State</ns2:Field.2.Name>
         <ns2:Field.2.Index>2</ns2:Field.2.Index>
         <ns2:Field.2.Value/>
         <ns2:Field.2.CommitFlag>N</ns2:Field.2.CommitFlag>
         <ns2:Field.3.Name>County</ns2:Field.3.Name>
         <ns2:Field.3.Index>3</ns2:Field.3.Index>
         <ns2:Field.3.Value/>
         <ns2:Field.3.CommitFlag>N</ns2:Field.3.CommitFlag>
         <ns2:Field.4.Name>Street</ns2:Field.4.Name>
         <ns2:Field.4.Index>4</ns2:Field.4.Index>
         <ns2:Field.4.Value/>
         <ns2:Field.4.CommitFlag>N</ns2:Field.4.CommitFlag>
         <ns2:Field.5.Name>Company</ns2:Field.5.Name>
         <ns2:Field.5.Index>5</ns2:Field.5.Index>
         <ns2:Field.5.Value/>
         <ns2:Field.5.CommitFlag>N</ns2:Field.5.CommitFlag>
         <ns2:Field.6.Name>Building</ns2:Field.6.Name>
         <ns2:Field.6.Index>6</ns2:Field.6.Index>
         <ns2:Field.6.Value/>
         <ns2:Field.6.CommitFlag>N</ns2:Field.6.CommitFlag>
         <ns2:Field.7.Name>Premise-No</ns2:Field.7.Name>
         <ns2:Field.7.Index>7</ns2:Field.7.Index>
         <ns2:Field.7.Value/>
         <ns2:Field.7.CommitFlag>N</ns2:Field.7.CommitFlag>
         <ns2:Field.8.Name>Sub-Building</ns2:Field.8.Name>
         <ns2:Field.8.Index>8</ns2:Field.8.Index>
         <ns2:Field.8.Value/>
         <ns2:Field.8.CommitFlag>N</ns2:Field.8.CommitFlag>
         <ns2:Field.9.Name>PO-Box</ns2:Field.9.Name>
         <ns2:Field.9.Index>9</ns2:Field.9.Index>
         <ns2:Field.9.Value/>
         <ns2:Field.9.CommitFlag>N</ns2:Field.9.CommitFlag>
         <ns2:Field.10.Name>+4</ns2:Field.10.Name>
         <ns2:Field.10.Index>10</ns2:Field.10.Index>
         <ns2:Field.10.Value/>
         <ns2:Field.10.CommitFlag>N</ns2:Field.10.CommitFlag>
         <ns2:SearchFieldIndex>-1</ns2:SearchFieldIndex>
         <ns2:Alternatives.InContext/>
         <ns2:Alternatives.InContext.Count>
              0
         </ns2:Alternatives.InContext.Count>
         <ns2:Alternatives.OutContext/>
         <ns2:Alternatives.OutContext.Count>
              0
         </ns2:Alternatives.OutContext.Count>
         <ns2:user_fields/>
      </ns2:Result>
   </ns2:output_port>
</ns2:xml.BuildGlobalAddressResponse>
```

Using BuildGlobalAddress

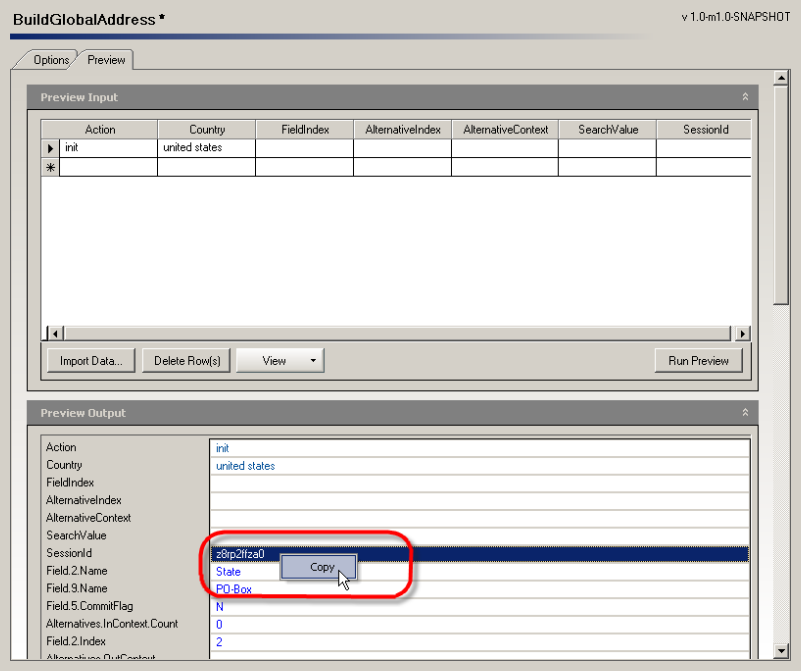
Building an address is an interactive process that requires you to select address elements at each step of the address building process. This means that building an address requires a sequence of calls to BuildGlobalAddress, not a single call. To start, you make an initialization call to BuildGlobalAddress. This call returns a session ID. You then use this session ID in subsequent calls. With each call, BuildGlobalAddress presents a list of alternative values for an address element. You select the value you want, then move on to the next address element until the complete address is built. With some exceptions, you need to make a separate call for each address element.

The overall process works like this:

- First, you make an initialization call to open a session and receive a system-assigned session ID.
- Make a search call to find possible values for a given address element.
- When you have selected the value you want, you make a commit call to indicate the value you want for the given address element.
- Continue to make search/commit calls until all address elements are committed.
- Finally, you make a close call to end the session.

To familiarize yourself with how the process works, use the Management Console's Preview tab to step through the following procedure.

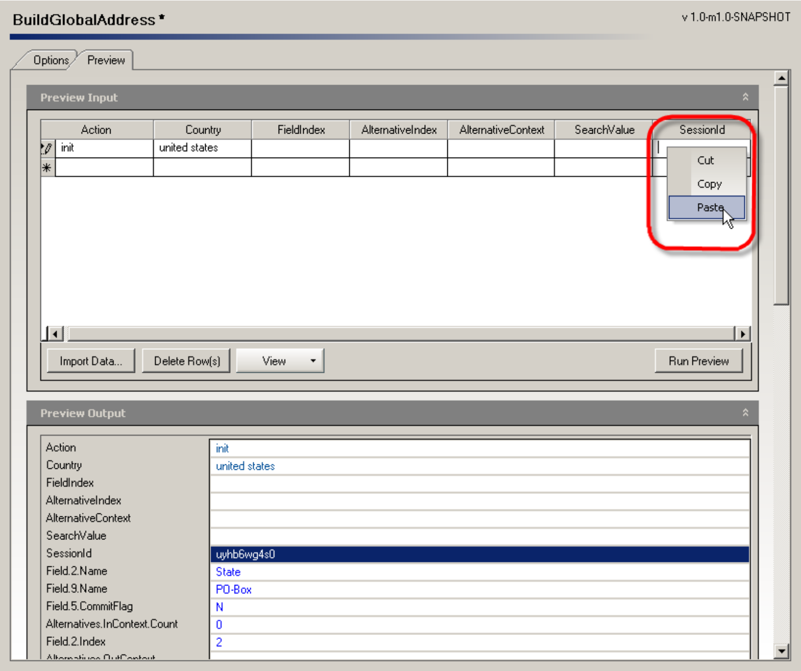
1. Open the Management Console.
2. Under the Services node, select **Build Global Address**.
3. On the **Options** tab, specify the options you want. For information on the options, see **Parameters for Options** on page 22.
4. Click the **Preview** tab.
5. In the **Action** field type `init`.
6. In the **Country** field enter the country of the address you want to build.
7. Click **Run Preview**.
8. Under Preview Output, find the **SessionId** field, right-click the value, and select `Copy` from the pop-up menu.



Action	Country	FieldIndex	AlternativeIndex	AlternativeContext	SearchValue	SessionId
init	united states					

Action	init
Country	united states
FieldIndex	
AlternativeIndex	
AlternativeContext	
SearchValue	
SessionId	z8rp2lfza0
Field.2.Name	State
Field.9.Name	PO-Box
Field.5.CommitFlag	N
Alternatives.InContext.Count	0
Field.2.Index	2
Alternatives.OutContext	

9. Under Preview Input, right-click the **SessionId** field and select `Paste`.



Action	Country	FieldIndex	AlternativeIndex	AlternativeContext	SearchValue	SessionId
init	united states					

Action	init
Country	united states
FieldIndex	
AlternativeIndex	
AlternativeContext	
SearchValue	
SessionId	uyhb6wg4s0
Field.2.Name	State
Field.9.Name	PO-Box
Field.5.CommitFlag	N
Alternatives.InContext.Count	0
Field.2.Index	2
Alternatives.OutContext	

10. Enter the following values in the input fields:

- Action—Type search.
- Country—Keep this field the same.
- FieldIndex—Type the index value of the first field you want to search. For example, if you know you want to search for an address in Chicago, you would type "1" because for U.S. addresses, field index 1 corresponds to the City field.
- SearchValue—Type the value you want to search for. For example, if you want to build an address in Chicago, you would type "chicago".
- SessionId—Keep the same value.

Note: The values in the other input fields are ignored.

11. Click **Run Preview** again.
12. The results of the search are placed in up to two output fields: **Alternatives.InContext** and **AlternativesOutContext**. For an explanation of the difference between in context results and out of context results, see **What Is Context?** on page 27.
13. When you have found the value you want, enter the following values in the input fields:

    - **Action**—Type **commit**.
    - **AlternativeIndex**—Type the index number for the alternative you choose. Index values start with 0, not 1. For example, if you search for Chicago the alternatives returned by BuildGlobalAddress would be indexed as follows. If you want to commit the value "CHICAGO" you would type "0" in the AlternativeIndex field.
        - 0—CHICAGO
        - 1—CHICAGO HTS
        - 2—CHICAGO PARK
        - 3—CHICAGO RIDGE
        - 4—EAST CHICAGO
        - 5—NORTH CHICAGO
        - 6—WEST CHICAGO
    - **AlternativeContext**—Type `in` or `out` to indicate whether the index value you specified in **AlternativeIndex** is for the list of alternatives in the **Alternatives.InContext** field or the **Alternatives.OutContext** field.
    - **SessionId**—Keep this value the same.

    Note: The values in the other input fields are ignored.

14. Click **Run Preview** again. The value you specified will now be in the Field.n.Value field for the appropriate address element.
15. Repeat the search and commit steps as often as needed until you have built the address.
16. Close the session by entering the following values in the input fields:

    - **Action**—Type **close**.
    - **SessionId**—Keep this value the same.

    Note: The values in the other input fields are ignored.

Request

Parameters for Input Data

Table 1: BuildGlobalAddress Input

Parameter	Format	Description
Data.Action	String	Specifies the action to take. One of the following: **init** Initialization. This action opens a session and returns a session ID which is required for all other actions. The init action requires the Country input field. **search** Searches for values for a specific address element and returns a list of alternative values for you to choose. The search action requires the following input fields:

Parameter	Format	Description
		• FieldIndex • SearchValue • SessionId **commit** Assigns one of the values returned by the search action to the field. The commit action requires the following input fields: • AlternativeIndex • AlternativeContext • SessionId **clear** Un-commits the field specified in the FieldIndex field. The clear action requires the following input fields: • FieldIndex • SessionID **close** Ends a session. The close action requires the SessionId input field.
Data.AlternativeContext	String	For the commit action, indicates whether you are choosing a value from the **Alternatives.InContext** field or the **Alternatives.OutContext** field. This field is ignored for other actions. One of the following: **in** You are committing a value from the **Alternatives.InContext** field. This means that the value you specify in the **AlternativeIndex** input field corresponds to a value in the **Alternatives.InContext** output field. **out** You are committing a value from the **Alternatives.OutContext** field. This means that the value you specify in the **AlternativeIndex** input field corresponds to a value in the **Alternatives.OutContext** output field.
Data.AlternativeIndex	String [79]	For the commit action, specifies the value you want to use in the address you are building. For example, if you searched for a city and BuildGlobalAddress returns a list of three cities, you would indicate the city you want by specifying the index value for your choice. Index values for the alternatives presented by BuildGlobalAddress are zero-based, meaning that the first alternative has an index of 0, the second alternative has a value of 1, and so on. The input field is ignored for actions other than commit.
Data.Country	String [79]	For the init action, specifies the country in which you want to build an address. Specify the country using the format you chose for input country format (English name two-character ISO 3116-1 Alpha-2 code, or three-character ISO 3116-1 Alpha-3 code). For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.

Parameter	Format	Description
		This input field is ignored for actions other than init.
Data.FieldIndex	String [79]	For the search action, specifies the address element that you want to search on. For the clear action, specifies the address element you want to un-commit. One of the following: **all** performs the "clear" action on all address elements. This option applies to the "clear" action only. **<IndexNumber>** Performs the action on a specific address element. To determine the index of an address element, first look at the Field.n.Name fields and locate the field you want. The value n indicates the field's index. For example, you want to look up ZIP Codes for U.S. addresses. After the init call you see that Field.0.Name is "Zip" indicating that the ZIP Code has a field index of "0". This input field is ignored for actions other than search and clear.
Data.SearchValue	String [79]	For the search action, specifies the value you want to search for. This value must be appropriate for the field you specified in FieldIndex. For example, if you specified the ZIP Code field in FieldIndex, then you would enter a ZIP Code or partial ZIP Code in this field. Likewise if you chose the city field in FieldIndex you would specify a city name or partial city name in this field. if you leave the field blank the search will return all values that are in context. For more information about in context and out of context values, see **What Is Context?** on page 27. This input field is ignored for actions other than search.
Data.SessionId	String [79]	Specifies the session ID you want to use for this call. To obtain a session ID use the init action. If a session is inactive for 5 minutes it will expire and you will need to perform a new init call to start a new session. This field is required for all actions except init.

Parameters for Options

Table 2: BuildGlobalAddress Options

Parameter	Description
Option.HomeCountry	Specifies the default country. You should specify the country where most of the addresses in your data are located. For example, if most of your addresses are in Canada, specify Canada. BuildGlobalAddress uses the country you specify to attempt validation when it cannot determine the country from the StateProvince, PostalCode, and Country address fields.

Parameter	Description
Option.OutputCountryFormat	Specifies the format to use for the country name in the output. One of the following: **E** The country in the output in English (default). **I** The country in the output as the two-character ISO code. **U** The country in the output as the three-character UPU code.
Option.OutputPostalCodeSeparator	Specifies whether or not to use separators (spaces or hyphens) in ZIP Codes or Canadian postal codes. For example, a ZIP + 4$^{®}$ Code with the separator would be 20706-1844 and without the separator it would be 207061844. A Canadian postal code with the separator would be P5E"1S7 and without the separator it would be P5E1S7. **Y** Yes, use separator (default). **N** No, do not use separator. **Note:** Spaces are used in Canadian postal codes and hyphens in U.S. ZIP + 4$^{®}$ Codes.
Option.ShowExtraAddressLine	Specifies whether or not to include the city, state/province, and postal code in one of the AddressLine output fields. Regardless of what you specify with this option, the output fields City, State/Province, and PostalCode will always contain the city, state/province, and postal code. **Y** Yes, include city, state/province, and postal code in an AddressLine output field (default). **N** No, do not include city, state/province, and postal code in an AddressLine output field.
Option.MaximumResults	Allows you to set the default value for this option to any value from 1 to 10000; it has a default value of 50 records. Note that values set in Enterprise Designer override those set in Management Console

Response

Address Data

Table 3: BuildGlobalAddress Output

Response Element	Format	Description
Action	String [79]	Shows the value specified in the Action input field for this call. For more information on this input field see **Parameters for Input Data** on page 20.
AddressLine1	String [79]	The formatted first address line.
AddressLine2	String [79]	The formatted second address line.
AddressLine3	String [79]	The formatted third address line.
AddressLine4	String [79]	The formatted fourth address line.
AddressLine5	String [79]	The formatted fifth address line.
AddressLine6	String [79]	The formatted sixth address line.

Response Element	Format	Description
AddressLine7	String [79]	The formatted seventh address line.
AddressLine8	String [79]	The formatted eighth address line.
AlternativeContext	String [79]	Shows the value specified in the AlternativeContext input field for this call. For more information, see **Parameters for Input Data** on page 20.
AlternativeIndex	String [79]	Shows the value specified in the AlternativeIndex input field for this call. For more information on this input field see **Parameters for Input Data** on page 20.
Alternatives.InContext	String [79]	A comma-delimited list of the possible values for the field you searched on which fit the context of fields you have already committed. For information on context see **What Is Context?** on page 27.
Alternatives.InContext.Count	String [79]	The number of "in context" results returned by your search. For information on context see **What Is Context?** on page 27.
Alternatives.OutContext	String [79]	A comma-delimited list of the possible values for the field you searched on which do not fit the context of fields you have already committed. For information on context see **What Is Context?** on page 27.
Alternatives.OutContext.Count	String [79]	The number of "out of context" results returned by your search. For information about context, see **What Is Context?** on page 27.
ApartmentLabel	String [79]	Apartment designator (such as STE or APT). For example: 123 E Main St. APT 3
ApartmentNumber	String [79]	Apartment number. For example: 123 E Main St. APT **3**
Building	String [79]	The name of a building.
City	String [79]	The city name.
Country	String [79]	Shows the value specified in the Country input field for this call. For more information about this input field, see **Parameters for Input Data** on page 20.
Country	String [79]	The two- or three-character ISO code, or English name of the country. For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Department	String [79]	The name of a distinct part of anything arranged into divisions. For example, the Finance Department in a corporation.
Field.n.CommitFlag	String [79]	Indicates whether you have chosen a value for field n (i.e. "committed" a value). One of the following: **Y** Yes, the value of this field has been committed.

Response Element	Format	Description
		N No, the value of this field has not been committed.
Field.n.Index	String [79]	An index value used to refer to field n, where n is 0 though 10. For example, for U.S. addresses the index value of the ZIP field is "0".
Field.n.Name	String [79]	The name of the address element contained in field n, where n is 0 through 10. For example, for U.S. addresses Field.0.Name is ZIP.
Field.n.Value	String [79]	The value that has been committed to field n, where n is 0 through 10. This field is blank on the init call.
FieldIndex	String [79]	Shows the value specified in the FieldIndex input field for this call. For more information on this input field see **Parameters for Input Data** on page 20.
FirmName	String [79]	The name of a company. For example: **Pitney Bowes Software** 4200 PARLIAMENT PL STE 600 LANHAM MD 20706-1844 USA
HouseNumber	String [79]	House number. For example: **123** E Main St. Apt 3
POBox	String [79]	The post office box number. If the address is a rural route address, the rural route box number will appear here.
PostalCode	String [79]	The postal code. In the U.S. this is the ZIP Code™.
PostalCode.AddOn	String [79]	The 4-digit add-on part of the ZIP + 4® Code. For example, in the ZIP Code™ 60655-1844, 1844 is the 4-digit add-on. (U.S. addresses only.)
PostalCode.Base	String [79]	The 5-digit ZIP Code™. For example 20706 (U.S. addresses only.)
Principality	String [79]	An area within a country. For example, England, Scotland, and Wales are principalities. This field will normally be blank.
SearchFieldIndex	String [79]	The index value of the field searched in the previous search action.
SearchValue	String [79]	Shows the value specified in the SearchValue input field for this call. For more information on this input field see **Parameters for Input Data** on page 20.
SessionId	String [79]	Shows the value specified in the SessionId input field for this call. For more information on this input field see **Parameters for Input Data** on page 20.
StateProvince	String [79]	The state or province abbreviation.
StreetName	String [79]	Street name. For example: 123 E **Main** St. Apt 3

Response Element	Format	Description
StreetSuffix	String [79]	Street suffix. For example: 123 E Main **St**. Apt 3
SubCity	String [79]	A district or suburb. The subcity is used in countries where it is common to include the district or suburb within the address. For example, 27 Crystal Way **Bradley Stoke** Bristol BS32 8GA In this case, "Bradley Stoke" is the subcity.
SubStreet	String [79]	The second street address used to identify an address. Substreets are used in countries where it is common to give two street names in the address. For example, 12 The Mews **High Street** In this example, "High Street" is the substreet. Substreets can be used to precisely identify the delivery location. In the example, "The Mews" may be a small street that needs another street identification to properly locate the address, so "High Street" is included. In this case, "High Street" is the main or known street.
USCountyName	String [79]	For U.S. addresses, the name of the county where the address is located.

Return Codes

Table 4: BuildGlobalAddress Return Codes

Response Element	Format	Description
Status	String [79]	Reports the success or failure of the match attempt. **null** Success **F** Failure
Status.Code	String [79]	Reason for failure, if there is one. • SessionError • SeverError • CountryNotFound
Status.Description	String [79]	Description of the problem, if there is one. **Please initialize new session** This value will appear if Status.Code=SessionError. **Null or empty action** This value will appear if Status.Code=SessionError. **Unknown action** This value will appear if Status.Code=SessionError.

Response Element	Format	Description
		Invalid session This value will appear if Status.Code=SessionError.
		Invalid value for This value will appear if Status.Code=SessionError.
		Cannot Search Committed Field This value will appear if Status.Code=SessionError.
		Module not licensed This value will appear if Status.Code=ServerError.
		Could Not Identify Country This value will appear if Status.Code=CountryNotFound

What Is Context?

When you perform a search for an address element, BuildGlobalAddress looks at the address elements that you have already committed and splits up the values it returns based on whether or not the returned values exist within the context of the address elements you have already committed. For example, in the U.S. the following cities exist:

In Illinois:

- CHICAGO
- CHICAGO HTS
- CHICAGO RIDGE
- NORTH CHICAGO
- WEST CHICAGO

In Indiana:

- EAST CHICAGO

In Nevada:

- CHICAGO PARK

If you have already committed a value of "IN" (Indiana) for the state and then searched for the city "chicago", BuildGlobalAddress would return EAST CHICAGO as an "in context" result because it exists in Indiana, and it would return all the other matches for "chicago" as out-of-context results. Likewise, if you committed a value of "IL" (Illinois) for the state, BuildGlobalAddress would return EAST CHICAGO and CHICAGO PARK as out of context, and CHICAGO, CHICAGO HTS, CHICAGO RIDGE, NORTH CHICAGO, and WEST CHICAGO as "in context."

GetGlobalCandidateAddresses

GetGlobalCandidateAddresses returns a list of addresses that are considered matches for a given input address. If the input address matches multiple addresses in the Address Now database, the possible matches are returned. If the input address matches only one address in the Address Now database, no address data is returned.

GetGlobalCandidateAddresses is part of the Address Now Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/GetGlobalCandidateAddresses/results.json
```

XML endpoint:

```
http://server:port/rest/GetGlobalCandidateAddresses/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GetGlobalCandidateAddresses/results.json?
Data.AddressLine1=1070+Maple&Data.City=Batavia&Data.StateProvince=NY
```

The JSON returned by this request would be:

Note: Empty response elements have been removed from this example. Only the first two candidate address are shown.

```
{"output_port": [
      {
      "Confidence": "80",
      "AddressLine1": "1 Maple St",
      "AddressLine2": "Batavia NY 14020-3116",
      "HouseNumber": "1",
      "StreetName": "Maple",
      "StreetSuffix": "St",
      "City": "Batavia",
      "USCountyName": "Genesee",
      "StateProvince": "NY",
      "PostalCode": "14020-3116",
      "Country": "United States",
      "ACRCode": "L5-P5S5A0T4R4Z6C4-080",
      "PostalCode.Base": "14020",
      "PostalCode.AddOn": "3116",
      "user_fields": []
   },
      {
      "Confidence": "80",
      "AddressLine1": "3 Maple St",
      "AddressLine2": "Batavia NY 14020-3116",
      "HouseNumber": "3",
      "StreetName": "Maple",
      "StreetSuffix": "St",
      "City": "Batavia",
      "USCountyName": "Genesee",
      "StateProvince": "NY",
      "PostalCode": "14020-3116",
      "Country": "United States",
      "ACRCode": "L5-P5S5A0T4R4Z6C4-080",
      "PostalCode.Base": "14020",
      "PostalCode.AddOn": "3116",
      "user_fields": []
   }
]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GetGlobalCandidateAddresses/results.xml?
Data.AddressLine1=1070+Maple&Data.City=Batavia&Data.StateProvince=NY
```

The XML returned by this request would be:

Note: Empty response elements have been removed from this example. Only the first two candidate address are shown.

```
<ns2:xml.GetGlobalCandidateAddressesResponse
xmlns:ns2="http://www.pb.com/spectrum/services/GetGlobalCandidateAddresses">

   <ns2:output_port>
      <ns2:Address>
         <ns2:Confidence>80</ns2:Confidence>
         <ns2:AddressLine1>1 Maple St</ns2:AddressLine1>
         <ns2:AddressLine2>Batavia NY 14020-3116</ns2:AddressLine2>
         <ns2:HouseNumber>1</ns2:HouseNumber>
         <ns2:StreetName>Maple</ns2:StreetName>
         <ns2:StreetSuffix>St</ns2:StreetSuffix>
         <ns2:City>Batavia</ns2:City>
         <ns2:USCountyName>Genesee</ns2:USCountyName>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>14020-3116</ns2:PostalCode>
         <ns2:PostalCode.Base>14020</ns2:PostalCode.Base>
         <ns2:PostalCode.AddOn>3116</ns2:PostalCode.AddOn>
         <ns2:Country>United States</ns2:Country>
         <ns2:ACRCode>L5-P5S5A0T4R4Z6C4-080</ns2:ACRCode>
      </ns2:Address>
      <ns2:Address>
         <ns2:Confidence>80</ns2:Confidence>
         <ns2:AddressLine1>3 Maple St</ns2:AddressLine1>
         <ns2:AddressLine2>Batavia NY 14020-3116</ns2:AddressLine2>
         <ns2:HouseNumber>3</ns2:HouseNumber>
         <ns2:StreetName>Maple</ns2:StreetName>
         <ns2:StreetSuffix>St</ns2:StreetSuffix>
         <ns2:City>Batavia</ns2:City>
         <ns2:USCountyName>Genesee</ns2:USCountyName>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>14020-3116</ns2:PostalCode>
         <ns2:PostalCode.Base>14020</ns2:PostalCode.Base>
         <ns2:PostalCode.AddOn>3116</ns2:PostalCode.AddOn>
         <ns2:Country>United States</ns2:Country>
         <ns2:ACRCode>L5-P5S5A0T4R4Z6C4-080</ns2:ACRCode>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.GetGlobalCandidateAddressesResponse>
```

Request

Parameters for Input Data

GetGlobalCandidateAddresses takes a standard address as input. All addresses use this format no matter what country the address is from. AddressLine1 and Country are required input fields. The other fields are optional.

Table 5: GetGlobalCandidateAddresses Input

Parameter	Format	Description
Data.AddressLine1	String [79]	First address line. This is a required field.
Data.AddressLine2	String [79]	Second address line
Data.AddressLine3	String [79]	Third address line

Parameter	Format	Description
Data.AddressLine4	String [79]	Fourth address line
Data.AddressLine5	String [79]	Fifth address line
Data.AddressLine6	String [79]	Sixth address line
Data.AddressLine7	String [79]	Seventh address line
Data.AddressLine8	String [79]	Eighth address line
Data.City	String [79]	City name
Data.StateProvince	String [79]	State or province.
Data.PostalCode	String [10]	The postal code for the address in one of these formats: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999
Data.County	String	The country. Specify the country using the format you chose for input country format (English name or ISO code). For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Data.FirmName	String [79]	Company or firm name

Parameters for Options

Table 6: GetGlobalCandidateAddresses Options

Parameter	Description
Option.HomeCountry	Specifies the default country. Specify the country that is the destination of most of your mailpieces. For example, if most of your mailpieces are going to Canada, specify Canada. GetGlobalCandidateAddresses uses the country you specify to attempt validation when it cannot determine the country from the StateProvince, PostalCode, and Country address fields.
Option.OutputCountryFormat	Specifies the format to use for the country name in the output. One of the following: **E** The country in the output in English (default). **I** The country in the output in the two-character ISO code. **U** The country in the output in the three-character UPU code.

Parameter	Description
Option.OutputCasing	Specifies the casing of the output data. One of the following: **M** The output in mixed case (default). For example: 123 Main St Mytown FL 12345 **U** The output in upper case. For example: 123 MAIN ST MYTOWN FL 12345
Option.OutputPostalCodeSeparator	Specifies whether or not to use separators (spaces or hyphens) in ZIP™ Codes or Canadian postal codes. For example, a ZIP + 4® Code with the separator would be 20706-1844 and without the separator it would be 207061844. A Canadian postal code with the separator would be P5E"1S7 and without the separator it would be P5E1S7. **Y** Yes, use separator (default) **N** No, do not use separator **Note:** Spaces are used in Canadian postal codes and hyphens in U.S. ZIP + 4® Codes.
Option.ShowExtraAddressLine	Specifies whether or not to include the city, state/province, and postal code in one of the AddressLine output fields. Regardless of what you specify with this option, the output fields City, State/Province, and PostalCode will always contain the city, state/province, and postal code. **Y** Yes, include city, state/province, and postal code in an AddressLine output field (default). **N** No, do not include city, state/province, and postal code in an AddressLine output field.
Option.MaximumResults	The maximum number of candidate addresses to output. The default is 50. The maximum value is 100.
Option.ReturnUserData	Specifies whether or not to include in the output data from the input address that could not be validated. **Y** Yes, include input data that could not be validated. **N** No, do not include input data that could not be validated (default).

Response

Address Data

Table 7: GetGlobalCandidateAddresses Address Data Output

Response Element	Format	Description
AddressLine1	String [79]	The formatted first address line.
AddressLine2	String [79]	The formatted second address line.
AddressLine3	String [79]	The formatted third address line.

Response Element	Format	Description
AddressLine4	String [79]	The formatted fourth address line.
AddressLine5	String [79]	The formatted fifth address line.
AddressLine6	String [79]	The formatted sixth address line.
AddressLine7	String [79]	The formatted seventh address line.
AddressLine8	String [79]	The formatted eighth address line.
ApartmentLabel	String [79]	Apartment designator (such as STE or APT). For example: 123 E Main St. **APT** 3
ApartmentNumber	String [79]	Apartment number. For example: 123 E Main St. APT **3**
Building	String [79]	The name of a building.
City	String [79]	The city name.
Country	String [79]	The ISO code or English name of the country. For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Department	String [79]	The name of a distinct part of anything arranged into divisions. For example, the Finance Department in a corporation.
FirmName	String [79]	The name of a company. For example: **Pitney Bowes Software** 4200 PARLIAMENT PL STE 600 LANHAM MD 20706-1844 USA
HouseNumber	String [79]	House number. For example: **123** E Main St. Apt 3
POBox	String [79]	Post office box number. If the address is a rural route address, the rural route box number will appear here.
PostalCode	String [79]	The postal code as required by the local postal authority. For example, in the U.S. the postal code is the ZIP Code.
PostalCode.AddOn	String [79]	For U.S. addresses, the last four digits of the ZIP + 4® Code.
PostalCode.Base	String [79]	For U.S. addresses, the five-digit ZIP Code.
Principality	String [79]	An area within a country. For example, England, Scotland, and Wales are principalities. This field will normally be blank.
StateProvince	String [79]	The state or province abbreviation.

Response Element	Format	Description
StreetName	String [79]	Street name. For example: 123 E **Main** St. Apt 3
StreetSuffix	String [79]	Street suffix. For example: 123 E Main **St**. Apt 3
SubCity	String [79]	A district or suburb. The subcity is used in countries where it is common to include the district or suburb within the address. For example, 27 Crystal Way **Bradley Stoke** Bristol BS32 8GA In this case, "Bradley Stoke" is the subcity.
SubStreet	String [79]	The second street address used to identify an address. Substreets are used in countries where it is common to give two street names in the address. For example, 12 The Mews **High Street** In this example, "High Street" is the substreet. Substreets can be used to precisely identify the delivery location. In the example, "The Mews" may be a small street that needs another street identification to properly locate the address, so "High Street" is included. In this case, "High Street" is the main or known street.
USCountyName	String [79]	For U.S. addresses, the name of the county where the address is located.

Return Codes

Table 8: GetGlobalCandidateAddresses Return Codes

Response Element	Format	Description
ACRCode	String [79]	The Address Correction Result (ACR) code describes what data has been changed in each record. For information on what this code means, see **The ACR Code** on page 286.
Confidence	String [79]	The level of confidence assigned to the address being returned. Range is from zero (0) to 100; zero indicates failure, 100 indicates a very high level of confidence that the match results are correct.
Status	String [79]	Reports the success or failure of the match attempt. **Null** Success **F** Failure
Status.Code	String [79]	Reason for failure, if there is one. • RequestFailed • ServerError • CountryNotFound

Response Element	Format	Description
Status.Description	String [79]	Description of the problem, if there is one. **Maximum records cannot be set to 0. Minimum value should be 1** — This value will appear if Status.Code=RequestFailed. **Address Not Found** — This value will appear if Status.Code=RequestFailed. **Module not licensed** — This value will appear if Status.Code=ServerError. **Could Not Identify Country** — This value will appear if Status.Code=CountryNotFound.

ValidateGlobalAddress

ValidateGlobalAddress provides enhanced address standardization and validation for addresses outside the U.S. and Canada. ValidateGlobalAddress can also validate addresses in the U.S. and Canada but its strength is validation of addresses in other countries. If you need to validate addresses outside the U.S. and Canada, you should consider using ValidateGlobalAddress.

ValidateGlobalAddress is part of the Address Now Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/ValidateGlobalAddress/results.json
```

XML endpoint:

```
http://server:port/rest/ValidateGlobalAddress/results.xml
```

Resource URL

JSON endpoint:

```
http://server:port/rest/ValidateAddressGlobal/results.json
```

XML endpoint:

```
http://server:port/rest/ValidateAddressGlobal/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/ValidateGlobalAddress/results.json?
Data.AddressLine1=1825+Kramer+Ln&Data.City=Austin&Data.StateProvince=TX
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Confidence": "100",
   "AddressLine1": "1825 Kramer Ln",
   "AddressLine2": "Austin TX 78758",
   "AddressLine3": "",
   "AddressLine4": "",
   "AddressLine5": "",
   "AddressLine6": "",
   "AddressLine7": "",
   "AddressLine8": "",
   "Department": "",
   "Building": "",
   "SubStreet": "",
   "SubCity": "",
   "Principality": "",
   "City": "Austin",
   "StateProvince": "TX",
   "PostalCode": "78758",
   "Country": "United States",
   "FirmName": "",
   "HouseNumber": "1825",
   "StreetName": "Kramer",
   "StreetSuffix": "Ln",
   "ApartmentLabel": "",
   "POBox": "",
   "USCountyName": "Travis",
   "WCRCode": "",
   "URL1": "",
   "URL2": "",
   "Email1": "",
   "Email2": "",
   "ACRCode": "L5-P4S4A0T4R4Z6C4-100",
   "OuterMatchScore": "40",
   "Latitude": "",
   "Longitude": "",
   "ECRCode": "CoBoPeSeXoCeRoPo",
   "PostalCode.Base": "78758",
   "PostalCode.AddOn": "",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/ValidateGlobalAddress/results.xml?
Data.AddressLine1=1825+Kramer+Ln&Data.City=Austin&Data.StateProvince=TX
```

The XML returned by this request would be:

Note: Empty response elements have been removed from this example.

```
<ns2:xml.ValidateGlobalAddressResponse
xmlns:ns2="http://www.pb.com/spectrum/services/ValidateGlobalAddress">
   <ns2:output_port>
      <ns2:Address>
         <ns2:Confidence>100</ns2:Confidence>
         <ns2:AddressLine1>1825 Kramer Ln</ns2:AddressLine1>
         <ns2:AddressLine2>Austin TX 78758</ns2:AddressLine2>
         <ns2:City>Austin</ns2:City>
         <ns2:StateProvince>TX</ns2:StateProvince>
         <ns2:PostalCode>78758</ns2:PostalCode>
         <ns2:PostalCode.Base>78758</ns2:PostalCode.Base>
         <ns2:PostalCode.AddOn/>
         <ns2:Country>United States</ns2:Country>
         <ns2:HouseNumber>1825</ns2:HouseNumber>
         <ns2:StreetName>Kramer</ns2:StreetName>
         <ns2:StreetSuffix>Ln</ns2:StreetSuffix>
         <ns2:USCountyName>Travis</ns2:USCountyName>
         <ns2:ACRCode>L5-P4S4A0T4R4Z6C4-100</ns2:ACRCode>
         <ns2:OuterMatchScore>40</ns2:OuterMatchScore>
         <ns2:ECRCode>CoBoPeSeXoCeRoPo</ns2:ECRCode>
```

```
        </ns2:Address>
    </ns2:output_port>
</ns2:xml.ValidateGlobalAddressResponse>
```

Request

Parameters for Input Data

ValidateGlobalAddress takes a standard address as input. All addresses use this format no matter what country the address is from.

Table 9: ValidateGlobalAddress Input

Parameter	Format	Description
Data.AddressLine1	String [79]	First address line
Data.AddressLine2	String [79]	Second address line
Data.AddressLine3	String [79]	Third address line
Data.AddressLine4	String [79]	Fourth address line
Data.AddressLine5	String [79]	Fifth address line
Data.AddressLine5	String [79]	Sixth address line
Data.AddressLine7	String [79]	Seventh address line
Data.AddressLine7	String [79]	Eighth address line
Data.City	String [79]	City name
Data.StateProvince	String [79]	State or province.
Data.PostalCode	String [79]: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999	The postal code for the address. In the U.S. this is the ZIP Code™.
Data.Country	String [79]	Specify the country using the format you chose for input country format (English name or ISO code). For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.

Parameter	Format	Description
Data.FirmName	String [79]	Company or firm name

Parameters for Options

Input Data Options

Table 10: ValidateGlobalAddress Input Data Options

Parameter	Description
Option.HomeCountry	Specifies the default country. You should specify the country where most of the addresses are located. For example, if most of the addresses you process are in Canada, specify Canada. ValidateGlobalAddress uses the home country to attempt validation when it cannot determine the country from the StateProvince, PostalCode, and Country address fields. For a list of valid values, see **Country ISO Codes and Module Support** on page 292.

Output Data Options

Table 11: ValidateGlobalAddress Output Data Options

Parameter	Description
Option.OutputCountryFormat	Specifies the format to use for the country name in the output. One of the following: **E** The country in the output is English (default). **I** The country in the output in the two-character ISO code. **U** The country in the output in the three-character UPU code.
Option.OutputCasing	Specifies the casing of the output data. One of the following: **M** The output in mixed case (default). For example: 123 Main St Mytown FL 12345 **U** The output in upper case. For example: 123 MAIN ST MYTOWN FL 12345
Option.OutputPostalCodeSeparator	Specifies whether to use a separator (spaces or hyphens) in ZIP™ Codes or Canadian postal codes. For example, a ZIP + 4® Code with the separator would be 20706-1844 and without the separator it would be 207061844. A Canadian postal code with the separator would be P5E"1S7 and without the separator it would be P5E1S7. **Y** Yes, use separator (default) **N** No, do not use separator **Note:** Spaces are used in Canadian postal codes and hyphens in U.S. ZIP + 4® Codes.
Option.ShowExtraAddressLine	Specifies whether to include the city, state/province, and postal code in one of the AddressLine output fields. Regardless of what you specify

Parameter	Description
	with this option, the output fields City, State/Province, and PostalCode will always contain the city, state/province, and postal code. **Y** Yes, include city, state/province, and postal code in an AddressLine output field (default). **N** No, do not include city, state/province, and postal code in an AddressLine output field.
Option.StandardizeAddressOnFail	Specifies whether to return a standardized address when an address cannot be validated. The address is formatted using the preferred address format for the address's country. If this option is not selected, the output address component fields (StreetName, HouseNumber, etc.) are blank when address validation fails. **N** No, do not format failed addresses (default). **Y** Yes, standardize failed addresses.
Option.FormatOnFail	Specifies whether to return a formatted address when an address cannot be validated. The address is formatted using the preferred address format for the address's country. **Y** Yes, return a formatted address when an address cannot be validated. **N** No, do not return a formatted address when an address cannot be validated (default).
Option.ValidateAddress	Enables address validation. Address validation does the following: • Matches components to the relevant country's reference data • Corrects spelling errors • Adds missing components • Corrects or adds postal codes **Y** Yes, validate addresses (default). **N** No, do not validate addresses.
Option.FormatAddress	Formats the address components into the statutory postal or custom formats. **Y** Yes, format addresses (default). **N** No, do not format addresses.

Standardization Options

Table 12: ValidateGlobalAddress Standardization Options

Parameter	Description
Option.StandardizeComponentDepartment	Specifies whether or not to populate the Department field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponentFirmName	Specifies whether or not to populate the FirmName field when standardizing an address. **Y** Yes (default)

Parameter	Description
	N No
Option.StandardizeComponent.Building	Specifies whether or not to populate the Building field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.SubBuilding	Specifies whether or not to populate the SubBuilding field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.HouseNumber	Specifies whether or not to populate the HouseNumber field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.SubStreet	Specifies whether or not to populate the SubStreet field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.StreetName	Specifies whether or not to populate the StreetName field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.POBox	Specifies whether or not to populate the POBox field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.SubCity	Specifies whether or not to populate the SubCity field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.City	Specifies whether or not to populate the City field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.USCountyName	Specifies whether or not to populate the USCountyName field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.StateProvince	Specifies whether or not to populate the StateProvince field when standardizing an address. **Y** Yes (default)

Parameter	Description
	N No
Option.StandardizeComponent.Principality	Specifies whether or not to populate the Principality field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.PostalCode	Specifies whether or not to populate the PostalCode field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.Plus4	Specifies whether or not to populate the +4 field when standardizing an address. **Y** Yes (default) **N** No
Option.StandardizeComponent.Country	Specifies whether or not to populate the Country field when standardizing an address. **Y** Yes (default) **N** No
Option.ReportVulgarWords	Specifies whether or not to look for vulgar words. If this option is enabled, ValidateGlobalAddress returns a value in the WCRCode output field to indicate the results. **Y** Yes **N** No (default)
Option.FlagVulgarWords	Specifies whether or not to mark vulgar words in the output using the format ">VulgarWord<". **Y** Yes **N** No (default)
Option.DebugOutput	This option controls whether or not to include troubleshooting information in the output fields Email1, Email2, URL1, and URL2. **Y** Yes **N** No (default)

Validation Options

Table 13: ValidateGlobalAddress Validation Options

Parameter	Description
Option.ValidateComponent.Department	Specifies whether or not to include the Department field when validating an address. **Y** Yes **N** No (default)

Parameter	Description
Option.ValidateComponent.FirmName	Specifies whether or not to include the FirmName field when validating an address. **Y** Yes **N** No (default)
Option.ValidateComponent.Building	Specifies whether or not to include the Building field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.SubBuilding	Specifies whether or not to include the SubBuilding field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.HouseNumber	Specifies whether or not to include the HouseNumber field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.SubStreet	Specifies whether or not to include the SubStreet field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.StreetName	Specifies whether or not to include the StreetName field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.POBox	Specifies whether or not to include the POBox field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.SubCity	Specifies whether or not to include the SubCity field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.City	Specifies whether or not to include the City field when validating an address. **Y** Yes (default) **N** No

Parameter	Description
Option.ValidateComponent.USCountyName	Specifies whether or not to include the USCountyName field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.StateProvince	Specifies whether or not to include the StateProvince field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.Principality	Specifies whether or not to include the Principality field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.PostalCode	Specifies whether or not to include the PostalCode field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.Plus4	Specifies whether or not to include the +4 field when validating an address. **Y** Yes (default) **N** No
Option.ValidateComponent.Country	Specifies whether or not to include the Country field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.Department	Specifies whether or not to correct the Country field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.FirmName	Specifies whether or not to correct the FirmName field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.Building	Specifies whether or not to correct the Building field when validating an address. **Y** Yes (default) **N** No

Parameter	Description
Option.ForceUpdate.SubBuilding	Specifies whether or not to correct the SubBuilding field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.HouseNumber	Specifies whether or not to correct the HouseNumber field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.SubStreet	Specifies whether or not to correct the SubStreet field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.StreetName	Specifies whether or not to correct the StreetName field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.POBox	Specifies whether or not to correct the POBox field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.SubCity	Specifies whether or not to correct the SubCity field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.City	Specifies whether or not to correct the City field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.USCountyName	Specifies whether or not to correct the USCountyName field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.StateProvince	Specifies whether or not to correct the StateProvince field when validating an address. **Y** Yes (default) **N** No

Parameter	Description
Option.ForceUpdate.Principality	Specifies whether or not to correct the Principality field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.PostalCode	Specifies whether or not to correct the PostalCode field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.Plus4	Specifies whether or not to correct the +4 field when validating an address. **Y** Yes (default) **N** No
Option.ForceUpdate.Country	Specifies whether or not to correct the Country field when validating an address. **Y** Yes (default) **N** No
Option.ReplaceAlias.Department	Specifies whether or not to overwrite the Department field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.FirmName	Specifies whether or not to overwrite the FirmName field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.Building	Specifies whether or not to overwrite the Building field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.SubBuilding	Specifies whether or not to overwrite the SubBuilding field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.HouseNumber	Specifies whether or not to overwrite the HouseNumber field if an alias is found in the Address Now database. **Y** Yes **N** No (default)

Parameter	Description
Option.ReplaceAlias.SubStreet	Specifies whether or not to overwrite the SubStreet field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.StreetName	Specifies whether or not to overwrite the StreetName field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.POBox	Specifies whether or not to overwrite the POBox field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.SubCity	Specifies whether or not to overwrite the Subcity field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.City	Specifies whether or not to overwrite the City field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.USCountyName	Specifies whether or not to overwrite the USCountyName field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.StateProvince	Specifies whether or not to overwrite the StateProvince field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.Principality	Specifies whether or not to overwrite the Principality field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.PostalCode	Specifies whether or not to overwrite the PostalCode field if an alias is found in the Address Now database. **Y** Yes (default) **N** No

Parameter	Description
Option.ReplaceAlias.Plus4	Specifies whether or not to overwrite the +4 field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.ReplaceAlias.Country	Specifies whether or not to overwrite the Country field if an alias is found in the Address Now database. **Y** Yes **N** No (default)
Option.CautiousUpdate	This option, used in conjunction with the "Force Update", ensures that no major changes are made to the data during processing. **Y** Yes **N** No (default)
Option.CrossComponentMatch	Specifies whether or not to correct common address standardization and validation errors by performing cross-component matching. Cross-component matching checks for matches between data found in one field in the input data and another field in the Address Now database. **Y** Yes **N** No (default)
Option.UseReferenceDiacritics	Specifies whether or not ValidateGlobalAddress modifies the address to match the diacritics (accents, umlauts, etc.) in the postal database when the only changes to the address are the diacritics . One of the following: **Y** Yes (default) **N** No For example, if **UseReferenceDiacritics** is enabled, the following would occur: Input City: Chalon-Sur-Saône City in the postal database: CHALON SUR SAONE Output City: CHALON SUR SAONE Input City: ARTEMIVS'K City in the postal database: ARTEMIVSK Output City: ARTEMIVSK If **UseReferenceDiacritics** is not enabled, the following would occur: Input City: Chalon-Sur-Saône Reference City: CHALON SUR SAONE Output City: Chalon-Sur-Saône Input City: ARTEMIVS'K Reference City: ARTEMIVSK Output City: ARTEMIVS'K Note that this option has no effect on the **Transliteration** option.

Parameter	Description
Option.KeepStandardizationChanges	Specifies whether or not standardization changes such as changing "ROAD" to "RD" should be reported in the ACR code. **Y** Yes **N** No (default)
Option.AcceptanceLevel	The Acceptance Level setting specifies the minimum number of address components that must be validated in order for the whole address to be considered validated. The value specified for AcceptanceLevel corresponds to the second character of the ACR code. For more information, see **The ACR Code** on page 286. The acceptance level differs from the InnerMatchScore option in that acceptance level measures how many components Validate Global Address validated, regardless of how well the validated components matched to address components in the postal databases, whereas InnerMatchScore indicates the probability that the output address is the correct, validated version of the input address. One of the following: **-1** The acceptance level is automatically set to an appropriate level based on the address's country. For example, U.S. addresses are processed with an acceptance level of 4. **0** No components validated (default) **1** Country only validated **2** City and country validated **3** City, postal code and country validated **4** Street, city, postal code and country validated **5** Premise number, building name, sub-building, PO box, company, street, city, postal code, and country validated
Option.InnerMatchScore	Specifies the minimum confidence level for address validation. Addresses with a value in the Confidence output field greater than or equal to this value is validated, and those that have a lower value will not be validated (the output field Status will contain F.) Specify any value between 0 and 100. The higher the value, the higher the degree of confidence necessary for effective address validation. The default is 60.
Option.CompanyWeight	A whole number from 0 to 10, indicating the relative importance of the FirmName field compared to the data in the Address Now database. This affects the confidence value, and can be used to tailor the confidence to distinguish correct and incorrect updates. For more information, see **The ACR Code** on page 286. The default value is 1.
Option.StreetWeight	A whole number from 0 to 10, indicating the relative importance of the StreetName field compared to the data in the Address Now database. A whole number from 0 to 10, indicating the relative importance of this field compared to the others. For more information, see **The ACR Code** on page 286.

Parameter	Description
	The default value is 10.
Option.CityWeight	A whole number from 0 to 10, indicating the relative importance of the City field compared to the data in the Address Now database. A whole number from 0 to 10, indicating the relative importance of this field compared to the others. For more information, see **The ACR Code** on page 286. The default value is 8.
Option.PostcodeWeight	A whole number from 0 to 10, indicating the relative importance of the PostalCode field compared to the data in the Address Now database. A whole number from 0 to 10, indicating the relative importance of this field compared to the others. For more information, see **The ACR Code** on page 286. The default value is 8
Option.OuterMatchScoreLines	A value from 0 to 8 indicating the number of address lines to use when calculating the outer match score. The default is 8. For more information on the outer match score, see **The Outer Match Score** on page 54.

Output Format Options

Table 14: ValidateGlobalAddress Output Format Options

Parameter	Description
Option.FormatComponent.Department	Specifies whether or not the Department field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.FirmName	Specifies whether or not the FirmName field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.Building	Specifies whether or not the Building field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.SubBuilding	Specifies whether or not the SubBuilding field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.HouseNumber	Specifies whether or not the HouseNumber field should be included in the output of a formatted address. **Y** Yes (default) **N** No

Parameter	Description
Option.FormatComponent.SubStreet	Specifies whether or not the SubStreet field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.StreetName	Specifies whether or not the StreetName field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.POBox	Specifies whether or not the POBox field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.SubCity	Specifies whether or not the SubCity field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.City	Specifies whether or not the City field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.USCountyName	Specifies whether or not the USCountyName field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.StateProvince	Specifies whether or not the StateProvince field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.Principality	Specifies whether or not the Principality field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.PostalCode	Specifies whether or not the PostalCode field should be included in the output of a formatted address. **Y** Yes (default) **N** No
Option.FormatComponent.Plus4	Specifies whether or not the +4 field should be included in the output of a formatted address. **Y** Yes (default) **N** No

Parameter	Description
Option.FormatComponent.Country	Specifies whether or not the Country field should be included in the output of a formatted address. **Y** Yes **N** No (default)
Option.Transliteration	Specifies how to format diacritics in the output address. One of the following: **0** No transliteration is performed. Diacritic characters are left as specified in the input and/or postal database. Default. **1** Diacritic characters are removed and replaced with the equivalent unadorned character. **2** Diacritic characters are transliterated to an equivalent unadorned character or character sequence using language-specific transliteration rules. For example, the following shows the effect of each of the three transliteration options on a Swedish address. Note the differences in "Västra Frölunda". **0** Gustaf Wernersgata 12 S-42132 Västra Frölunda **1** Gustaf Wernersgata 12 S-42132 Vastra Frolunda **2** Gustaf Wernersgata 12 S-42132 Vaestra Froelunda

Response

Address Data Output

Table 15: ValidateGlobalAddress Address Data Output

Response Element	Format	Description
AddressLine1	String [79]	The formatted first address line.
AddressLine2	String [79]	The formatted second address line.
AddressLine3	String [79]	The formatted third address line.
AddressLine4	String [79]	The formatted fourth address line.
AddressLine5	String [79]	The formatted fifth address line.
AddressLine6	String [79]	The formatted sixth address line.
AddressLine7	String [79]	The formatted seventh address line.

Response Element	Format	Description
AddressLine8	String [79]	The formatted eighth address line.
ApartmentLabel	String [79]	Apartment designator (such as STE or APT). For example: 123 E Main St. APT 3
ApartmentNumber	String [79]	Apartment number. For example: 123 E Main St. APT **3**
Building	String [79]	The name of a building.
City	String [79]	The city name.
Country	String [79]	The ISO code or English name of the country. For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Department	String [79]	A subdivision of a country used in French and Spanish speaking countries. For example, France is divided into 100 departments.
FirmName	String [79]	The name of a company. For example: **Pitney Bowes Software** 4200 PARLIAMENT PL STE 600 LANHAM MD 20706-1844 USA
HouseNumber	String [79]	House number. For example: **123** E Main St. Apt 3
Latitude	String [79]	The most precise latitude that could be determined for the address. This could be a point level location or a centroid. The level of precision can be determined by looking at the ECRCode output field. For more information, see **The ECR Code** on page 53.
Longitude	String [79]	The most precise longitude that could be determined for the address. This could be a point level location or a centroid. The level of precision can be determined by looking at the ECRCode output field. For more information, see **The ECR Code** on page 53.
POBox	String [79]	The post office box number. If the address is a rural route address, the rural route box number will appear here.
PostalCode	String [79]	The postal code. In the U.S. this is the ZIP Code™.
PostalCode.AddOn	String [79]	The 4-digit add-on part of the ZIP + 4® Code. For example, in the ZIP Code™ 60655-1844, 1844 is the 4-digit add-on. (U.S. addresses only.)
PostalCode.Base	String [79]	The 5-digit ZIP Code™. For example 20706 (U.S. addresses only.)
Principality	String [79]	An area within a country. For example, England, Scotland, and Wales are principalities. This field will normally be blank.

Response Element	Format	Description
StateProvince	String [79]	The state or province abbreviation.
StreetName	String [79]	Street name. For example: 123 E **Main** St. Apt 3
StreetSuffix	String [79]	Street suffix. For example: 123 E Main **St**. Apt 3
SubCity	String [79]	A district or suburb. The subcity is used in countries where it is common to include the district or suburb within the address. For example, 27 Crystal Way **Bradley Stoke** Bristol BS32 8GA In this case, "Bradley Stoke" is the subcity.
SubStreet	String [79]	The second street address used to identify an address. Substreets are used in countries where it is common to give two street names in the address. For example, 12 The Mews **High Street** In this example, "High Street" is the substreet. Substreets can be used to precisely identify the delivery location. In the example, "The Mews" may be a small street that needs another street identification to properly locate the address, so "High Street" is included. In this case, "High Street" is the main or known street.
USCountyName	String [79]	For U.S. addresses, the name of the county where the address is located.

Return Codes

Table 16: ValidateGlobalAddress Return Codes

Response Element	Format	Description
ACRCode	String [79]	The Address Correction Result (ACR) code describes what data has been changed in each record. For information on what this code means, see **The ACR Code** on page 286.
Confidence	String [79]	The level of confidence assigned to the address being returned. Range is from zero (0) to 100; zero indicates failure, 100 indicates a very high level of confidence that the match results are correct. This value is the same as the last three digits of the ACR code, referred to as the validation match score. For more information, see **The ACR Code** on page 286.
ECRCode	String [79]	The Enhanced Correction Result (ECR) code describes the level of precision of the latitude and longitude returned for the address. For mroe information, see **The ECR Code** on page 53.

Response Element	Format	Description
Email1	String [79]	Extra standardization information.
Email2	String [79]	Extra standardization information.
OuterMatchScore	String [79]	A score that measures changes to each address line. For more information, see **The Outer Match Score** on page 54.
Status	String [79]	Reports the success or failure of the match attempt. • **null**—Success • **F**—Failure
Status.Code	String [79]	Reason for failure, if there is one. • UnableToValidate • ServerError • CountryNotFound
Status.Description	String [79]	Description of the problem, if there is one. • **Address Not Found**—This value will appear if Status.Code=UnableToValidate. • **Module not licensed**—This value will appear if Status.Code=ServerError. • **Could Not Identify Country**—This value will appear if Status.Code=CountryNotFound.
URL1	String [79]	Extra standardization information.
URL2	String [79]	Extra standardization information.
WCRCode	String [79]	The Word Correction Result (WCR) code describes vulgar words found in the input address. The code has two components: • **Location code**—One of the following: • **AB**—Indicates a vulgarity was found in the address. • **NB**—Indicates a vulgarity was found in the name. • **Count**—The number of vulgar words found in the location indicated by the location code. For example, AB2 indicates that two vulgar words were found in the input address.

The ECR Code

The Enhanced Correction Result (ECR) code describes the level of precision of the latitude/longitude coordinates returned for the address. The code consists of a prefix followed by a dash then the body of the code.

The prefix always begins with "EL" followed by a number from 1 to 5 indicating the overall level of precision:

- 5—Point geocode
- 4—Street centroid
- 3—Postcode centroid

- 2—City centroid
- 1—Region centroid

The body of the code identifies the components that were used to match the address to a geocode. Note that the body consists of letters and numerals. Here's what they mean:

- P—Premise/house number, building or PO box
- S—Street
- T—City
- R—Region/state
- Z—Postal code
- C—Country

There are only two numeric options in the body: 4 or 0

- 4—The component data was available to make the geocode-address match.
- 0—The component date was not available.

For example: EL4-P0S4T4R4Z4C4

In this example, the 0 following the P tells us that premise/house number data was not available to make this address match; everything from street to country, however, was used in the assignment.

The Outer Match Score

The outer match score indicates how much ValidateGlobalAddress changed each address line to validate the address. The score compares the address lines before standardization and after validation and formatting. This score is only generated if you set the option to a value greater than 0.

The outer match score is similar to the validation match score, which is part of the ACR code (see **The ACR Code** on page 286). The difference is that the outer match score measures any change to an address line, including formatting, whereas the validation match score measures only whether or not the data could be validated,

For example, take the following input address lines before processing:

Address Line 1: 5 camden cres
Address Line 2: bath
Address Line 3: uk

After processing the address lines are:

Address Line 1: 5 Camden Crescent
Address Line 2: Bath
Address Line 3: BA1 5HY
Address Line 4: United Kingdom

This has a validation match score of 84% and in outer match score of 23%.

The validation match score is high because the address components were fairly accurate before validation. The street name was valid except for casing and use of an abbreviation. The city and country were both valid. The only thing not correct was the postal code (in this case it was missing). Hence the relatively high validation match score of 84%.

The outer match score is low because after formatting, the address lines are considerably different from the input. In this case, Address Line 3 contained "uk" on input, and contains "BA1 5HY" on output. Line 4 was empty on input, and in populated on output. Address line 1 has also changed. The outer score is therefore quite low.

Enterprise Routing Module

GetTravelBoundary

GetTravelBoundary determines a drive or walk time or distance boundary from a location. This feature obtains polygons corresponding to an isochrone or isodistance calculation. An isochrone is a polygon or set of points representing an area that can be traversed in a network from a starting point in a given amount of time. An isodistance is a polygon or set of points representing the area that is a certain distance from the starting point. The Get Travel Boundary operation (also known as an iso definition) takes a starting point, a unit (linear or time), one or more costs and their associated tags as input and returns the resulting travel boundary. Cost refers to the amount of time or distance to use in calculating an iso. A tag is a string that identifies the cost and is used to match the corresponding result. Multiple costs can be given as input by providing the costs as a ";" delimited string.

GetTravelBoundary is part of the Enterprise Routing Module.

Note: GetTravelBoundary is only available as a web service. The Get Travel Boundary is not available through the Java, C++, C, .NET, or COM APIs.

Resource URL

JSON endpoint:

```
http://server:port/rest/GetTravelBoundary/results.json
```

XML endpoint:

```
http://server:port/rest/GetTravelBoundary/results.xml
```

```
http://server:port/soap/GetTravelBoundary
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GetTravelBoundary/results.json?
Data.Latitude=33.751748&Data.Longitude=-84.364014&Data.TravelBoundaryCost=10&
Data.TravelBondaryCostUnits=Kilometers
```

The JSON returned by this request would be:

Note: Some of the points have been removed from this example to shorten it.

```
{"output_port": [{
   "IsoNodeResponse": [],
   "IsoPolygonResponse":      {
      "srsName": "epsg:4326",
      "Polygon": [        {
         "srsName": "epsg:4326",
         "Exterior": {"LineString": [{"Pos":              [
                          {
                "X": -84.34868168466456,
                "Y": 33.68373169496257
             },
                          {
                "X": -84.36945064055561,
                "Y": 33.69293307108579
```

```
            },
                        {
               "X": -84.3694506405556,
               "Y": 33.69293307108579
            },
                        {
               "X": -84.3694506405556,
               "Y": 33.69303002973829
            },
                        {
               "X": -84.37104825254721,
               "Y": 33.69391558543121
            },
                        {
               "X": -84.37104825254721,
               "Y": 33.6936408692491
            },
                        {
               "X": -84.42163929894845,
               "Y": 33.716054477754355
            },
                        {
               "X": -84.4440058668311,
               "Y": 33.710741143596806
            },
                        {
               "X": -84.43921303085625,
               "Y": 33.72800947960886
            },
                        {
               "X": -84.45678676276404,
               "Y": 33.73376559161287
            },
                        {
               "X": -84.43921303085625,
               "Y": 33.73996448146335
            },
            ...
         ]}]}
      }]
   },
   "user_fields": [   {
      "name": "TravelBondaryCostUnits",
      "value": "Kilometers"
   }]
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GetTravelBoundary/results.xml?
Data.TravelBoundaryCostUnits=Kilometers&Data.Latitude=33.751748&
Data.Longitude=-84.364014&Data.TravelBoundaryCost=10
```

The XML returned by this request would be:

Note: Some of the points have been removed from this example to shorten it.

```
<ns3:xml.GetTravelBoundaryResponse
xmlns:ns2="http://www.mapinfo.com/midev/service/geometries/v1"
xmlns:ns3="http://www.pb.com/spectrum/services/GetTravelBoundary">
   <ns3:output_port>
      <ns3:IsoRouteResponse>
         <ns3:IsoNodeResponse/>
         <ns3:IsoPolygonResponse
              xsi:type="ns2:MultiPolygon"
              srsName="epsg:4326"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <ns2:Polygon srsName="epsg:4326">
               <ns2:Exterior>
                  <ns2:LineString>
                     <ns2:Pos>
                        <ns2:X>-84.34868168466456</ns2:X>
                        <ns2:Y>33.68373169496257</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.36945064055561</ns2:X>
                        <ns2:Y>33.69293307108579</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.3694506405556</ns2:X>
                        <ns2:Y>33.69293307108579</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.3694506405556</ns2:X>
                        <ns2:Y>33.69303002973829</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.37104825254721</ns2:X>
                        <ns2:Y>33.69391558543121</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.37104825254721</ns2:X>
                        <ns2:Y>33.6936408692491</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.42163929894845</ns2:X>
                        <ns2:Y>33.716054477754355</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.4440058668311</ns2:X>
                        <ns2:Y>33.710741143596806</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.43921303085625</ns2:X>
                        <ns2:Y>33.72800947960886</ns2:Y>
                     </ns2:Pos>
                     <ns2:Pos>
                        <ns2:X>-84.45678676276404</ns2:X>
                        <ns2:Y>33.73376559161287</ns2:Y>
                     </ns2:Pos>
                     ...
                  </ns2:LineString>
               </ns2:Exterior>
            </ns2:Polygon>
         </ns3:IsoPolygonResponse>
         <ns3:user_fields/>
      </ns3:IsoRouteResponse>
   </ns3:output_port>
</ns3:xml.GetTravelBoundaryResponse>
```

Example

Case 1, Single Cost:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetTravelBoundary">
   <soapenv:Header/>
   <soapenv:Body>
      <get:GetTravelBoundaryRequest>
         <get:input_port>
            <get:IsoRouteRequest>
               <get:Latitude>33.751748</get:Latitude>
               <get:Longitude>-84.364014</get:Longitude>
               <get:TravelBoundaryCost>10</get:TravelBoundaryCost>
```

```
<get:TravelBoundaryCostUnits>Kilometers</get:TravelBoundaryCostUnits>
            </get:IsoRouteRequest>
         </get:input_port>
      </get:GetTravelBoundaryRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

This would be the response:

Note: Some of the points have been removed from this example to shorten it.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns4:GetTravelBoundaryResponse xmlns:ns2="http://spectrum.pb.com/"
      xmlns:ns3="http://www.mapinfo.com/midev/service/geometries/v1"
      xmlns:ns4="http://www.pb.com/spectrum/services/GetTravelBoundary">
         <ns4:output_port>
            <ns4:IsoRouteResponse>
               <ns4:IsoNodeResponse/>
               <ns4:IsoPolygonResponse
                    xsi:type="ns3:MultiPolygon"
                    srsName="epsg:4326"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

                  <ns3:Polygon srsName="epsg:4326">
                     <ns3:Exterior>
                        <ns3:LineString>
                           <ns3:Pos>
                              <ns3:X>-84.34868168466456</ns3:X>
                              <ns3:Y>33.68373169496257</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.36945064055561</ns3:X>
                              <ns3:Y>33.69293307108579</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.3694506405556</ns3:X>
                              <ns3:Y>33.69293307108579</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.3694506405556</ns3:X>
                              <ns3:Y>33.69303002973829</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.37104825254721</ns3:X>
                              <ns3:Y>33.69391558543121</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.37104825254721</ns3:X>
                              <ns3:Y>33.6936408692491</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.42163929894845</ns3:X>
                              <ns3:Y>33.716054477754355</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.4440058668311</ns3:X>
                              <ns3:Y>33.710741143596806</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.43921303085625</ns3:X>
                              <ns3:Y>33.72800947960886</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.45678676276404</ns3:X>
                              <ns3:Y>33.73376559161287</ns3:Y>
                           </ns3:Pos>
                           ...
                        </ns3:LineString>
                     </ns3:Exterior>
                  </ns3:Polygon>
               </ns4:IsoPolygonResponse>
               <ns4:user_fields/>
            </ns4:IsoRouteResponse>
         </ns4:output_port>
      </ns4:GetTravelBoundaryResponse>
   </soap:Body>
</soap:Envelope>
```

Case 2, Multiple Costs:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetTravelBoundary">
   <soapenv:Header/>
   <soapenv:Body>
      <get:GetTravelBoundaryRequest>
         <get:input_port>
            <get:IsoRouteRequest>
               <get:Latitude>33.751748</get:Latitude>
               <get:Longitude>-84.364014</get:Longitude>
               <get:TravelBoundaryCost>5;10</get:TravelBoundaryCost>

<get:TravelBoundaryCostUnits>Kilometers</get:TravelBoundaryCostUnits>
            </get:IsoRouteRequest>
         </get:input_port>
      </get:GetTravelBoundaryRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

This would be the response:

Note: Some of the points have been removed from this example to shorten it.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns4:GetTravelBoundaryResponse xmlns:ns2="http://spectrum.pb.com/"
      xmlns:ns3="http://www.mapinfo.com/midev/service/geometries/v1"
      xmlns:ns4="http://www.pb.com/spectrum/services/GetTravelBoundary">
         <ns4:output_port>
            <ns4:IsoRouteResponse>
      <ns4:cost>5</ns4:cost>
      <ns4:costUnits>Kilometers</ns4: costUnits >
      <ns4:IsoNodeResponse/>
               <ns4:IsoPolygonResponse
                    xsi:type="ns3:MultiPolygon"
                    srsName="epsg:4326"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

                  <ns3:Polygon srsName="epsg:4326">
                     <ns3:Exterior>
                        <ns3:LineString>
                           <ns3:Pos>
                              <ns3:X>-84.34868168466456</ns3:X>
                              <ns3:Y>33.68373169496257</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.36945064055561</ns3:X>
                              <ns3:Y>33.69293307108579</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.3694506405556</ns3:X>
                              <ns3:Y>33.69293307108579</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.3694506405556</ns3:X>
                              <ns3:Y>33.69303002973829</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.37104825254721</ns3:X>
                              <ns3:Y>33.69391558543121</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.37104825254721</ns3:X>
                              <ns3:Y>33.6936408692491</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.42163929894845</ns3:X>
                              <ns3:Y>33.716054477754355</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.4440058668311</ns3:X>
                              <ns3:Y>33.710741143596806</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.43921303085625</ns3:X>
                              <ns3:Y>33.72800947960886</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.45678676276404</ns3:X>
                              <ns3:Y>33.73376559161287</ns3:Y>
                           </ns3:Pos>
                           ...
                        </ns3:LineString>
                     </ns3:Exterior>
                  </ns3:Polygon>
               </ns4:IsoPolygonResponse>
               <ns4:user_fields/>
            </ns4:IsoRouteResponse>
   <ns4:IsoRouteResponse>
      <ns4:cost>10</ns4:cost>
      <ns4:costUnits>Kilometers</ns4: costUnits >
      <ns4:IsoNodeResponse/>
               <ns4:IsoPolygonResponse
                    xsi:type="ns3:MultiPolygon"
                    srsName="epsg:4326"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

                  <ns3:Polygon srsName="epsg:4326">
                     <ns3:Exterior>
                        <ns3:LineString>
                           <ns3:Pos>
                              <ns3:X>-84.34868168466456</ns3:X>
                              <ns3:Y>33.68373169496257</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.36945064055561</ns3:X>
                              <ns3:Y>33.69293307108579</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.3694506405556</ns3:X>
                              <ns3:Y>33.69293307108579</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.3694506405556</ns3:X>
                              <ns3:Y>33.69303002973829</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.37104825254721</ns3:X>
                              <ns3:Y>33.69391558543121</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.37104825254721</ns3:X>
                              <ns3:Y>33.6936408692491</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.42163929894845</ns3:X>
                              <ns3:Y>33.716054477754355</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.4440058668311</ns3:X>
```

```
                              <ns3:Y>33.710741143596806</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.43921303085625</ns3:X>
                              <ns3:Y>33.72800947960886</ns3:Y>
                           </ns3:Pos>
                           <ns3:Pos>
                              <ns3:X>-84.45678676276404</ns3:X>
                              <ns3:Y>33.73376559161287</ns3:Y>
                           </ns3:Pos>
                           ...
                        </ns3:LineString>
                     </ns3:Exterior>
                  </ns3:Polygon>
               </ns4:IsoPolygonResponse>
               <ns4:user_fields/>
            </ns4:IsoRouteResponse>
         </ns4:output_port>
      </ns4:GetTravelBoundaryResponse>
   </soap:Body>
</soap:Envelope>
```

Request

Parameters for Input Data

GetTravelBoundary takes cost, cost unit, point latitude, and point longitude as input. The following table provides information on the format and layout of the input.

Table 17: GetTravelBoundary Input Data

Parameter	Format	Description
Data.Latitude	String	Latitude of the point. Specify latitude in the format chosen in the Option.CoordinateFormat parameter.
Data.Longitude	String	Longitude of the point. Specify longitude in the format chosen in the Option.CoordinateFormat parameter.
Data.TravelBoundaryCost	String	(Optional) The cost distance or time, in the units specified by either the Data.TravelBoundaryCostUnits parameter or the Option.DefaultTravelBoundaryCostUnits parameter. For example, if the unit specified is miles and you specify 10 in this field, the cost would be 10 miles. Use this field to override the default travel boundary cost on a record-by-record basis. You can also specify multiple costs by specifying the values as a ";" delimited string. It will return a separate Iso Route Response for every cost specified. If you specify multiple costs, every response will have cost and costUnits associated with that response.
Data.TravelBoundaryCostUnits	String	(Optional) The type of metric used to calculate the travel boundary. One of the following: • Feet • Hours

Parameter	Format	Description
		• Kilometers • Meters • Miles • Minutes • Seconds • Yards Use this field to override the default travel boundary cost units on a record-by-record basis.

Parameters for Options

Input Options

Table 18: GetTravelBoundary Input Options

Parameter	Description
Option.DataSetResourceName	The name of the database that contains the data to use in the search process. Use the database name specified in the Enterprise Routing Module Routing Database Resource tool. For more information, see the *Spectrum™ Technology Platform Administration Guide*.
Option.CoordinateSystem	The coordinate system of the input coordinates. The format must be the European Petroleum Survey Group (EPSG) code, the SRID code, or the MAPINFO code. Specify the coordinate reference system in the format `codespace:code`. For example, a geometry in Latitude/Longitude (WGS 84) would be `EPSG:4326`. The default value is EPSG:4326, which is also known as Latitude/Longitude (WGS 84). This is a worldwide system and is the terrestrial reference frame associated with the NAVSTAR Global Positioning System (GPS) used extensively for navigation and surveying. This datum is earth-centered ("geocentric") and defined with satellite and terrestrial data. For more information on EPSG codes, see **www.spatialreference.org**. To retrieve a list of supported codespaces for EPSG, you can submit the SOAP request List Supported CoordSys by Code Space from the Geometry Service Demo page at **http://localhost:8080/Spatial/GeometryService/DemoPage.html**.
Option.CoordinateFormat	Specifies the format of latitude/longitude coordinates in the input. **Note:** Use this option only if you specify a Latitude/Longitude coordinate system. If the coordinate system is not a Latitude/Longitude coordinate system, set the coordinate format to Decimal. One of the following: **Decimal** (90.000000, 180.000000) **DecimalAssumed** (90000000, 180000000). Default. **DegreesMinutesSeconds** (90 00 00N, 180 00 00W) **PreZero** (090000000N, 180000000W) **PreZeroDecimal** (090.000000N, 180.000000W)

Parameter	Description
Option.DefaultTravelBoundaryCost	Number of cost units. The default is 10.
Option.DefaultTravelBoundaryCostUnits	Type of metric you want to use to calculate the travel boundary. One of the following: • Feet • Hours • Kilometers • Meters • Miles • Minutes • Seconds • Yards
Option.historicTrafficTimeBucket	Specifies whether the routing calculation uses the historic traffic speeds. These speeds are based on different time-of-day buckets. The data must have historic traffic speeds included in order to use this feature. The data for each country/region has the same bucket definitions, where the speeds for these bucket values may vary. The options are: **None** The default value. Historic traffic data is not used in the calculation. Instead an averaged speed value is used. **AMPeak** Calculate routes with the peak AM speeds. The AMPeak time bucket is from 07:00 to 10:00hr time of day. **PMPeak** Calculate routes with the peak PM speeds. The PMPeak time bucket is from 16:00 to 19:00hr time of day. **OffPeak** Calculate routes with the off peak (daytime) speeds. The OffPeak time bucket is from 10:00 to 16:00hr time of day. **Night** Calculate routes with the nighttime speeds. The Night time bucket is from 22:00 to 04:00hr time of day.

Output Options

Table 19: GetTravelBoundary Output Options

Parameter	Description
Option.ResultType	Specifies the type of result you want returned. One of the following: **AccessibleNodes** Returns the entire isoChrone. **Geometry** Returns all of the points along the road network that can be reached for the isoChrone calculation. **StartNodes** Returns the location specified by the entered address.
Option.SimplificationFactor	Specifies what percentage of the original points should be returned or upon which the resulting polygon should be based.

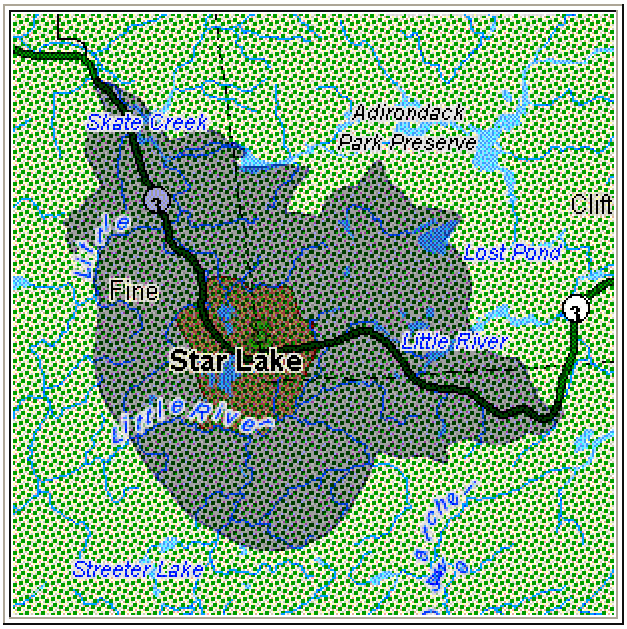
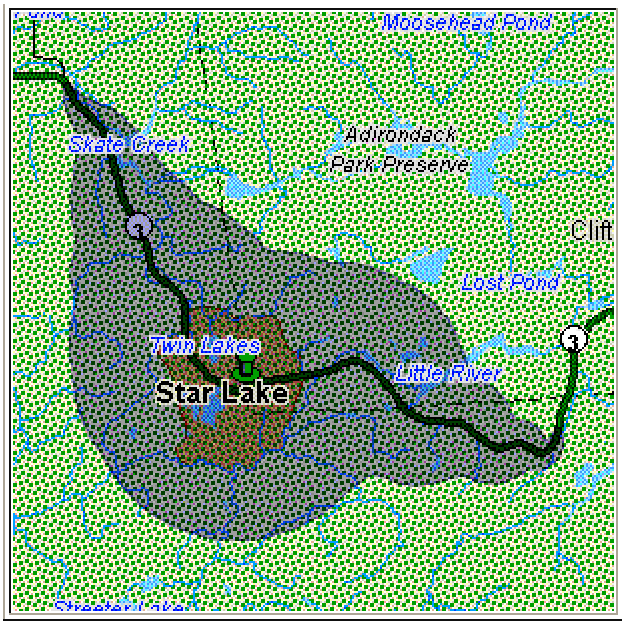
Parameter	Description
Option.BandingStyle	Specifies the style of banding to be used in the result. Banding styles are the types of multiple isoChrone or distance bands that can be displayed based on multiple costs. **Donut** Each boundary is determined by subtracting out the next smallest boundary. **Encompassing** Each boundary is determined independent of all others.
Option.ReturnHoles	Specifies whether you want to return holes, which are areas within the larger boundary that cannot be reached within the desired time or distance, based on the road network. **Y** Yes, return holes. **N** Do not return holes. Default.
Option.ReturnIslands	Specifies whether you want to return islands, which are small areas outside the main boundary that can be reached within the desired time or distance. **Y** Yes, return islands. **N** Do not return islands. Default.

Travel Options

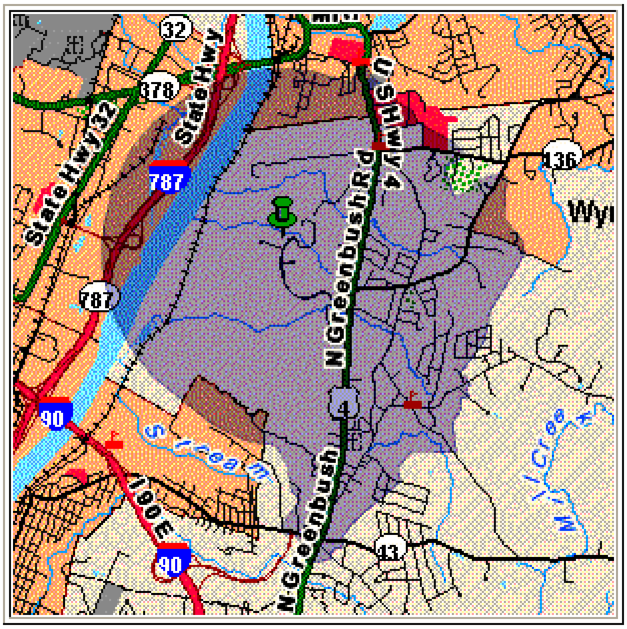
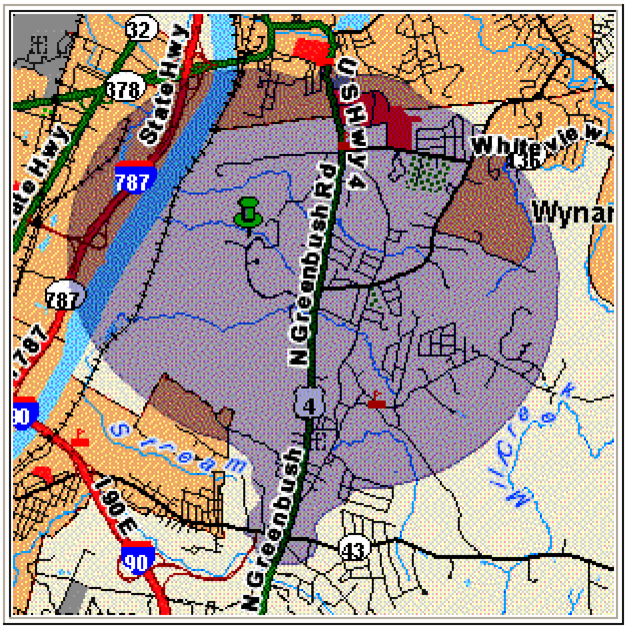
Travel options specify assumptions to make about travel speed off network roads and whether to use only major roads when calculating the travel boundary. Most travel options have to do with ambient speed.

Table 20: GetTravelBoundary Travel Options

Parameter	Description
Option.MaximumOffRoadDistance	Specifies the maximum distance to allow travel off the road network. Examples of off-network roads include driveways and access roads. For example, if you specify a maximum off road distance of 1 mile the travel boundary will extend no further than one mile from the network road. If you specify a value of 0 the travel boundary will not extend beyond the road itself. Use the ambient speed options to specify the speed of travel along non-network roads.
Option.Units	The units of measure in which you want the data returned. One of the following: • Kilometer (default) • Meter • Mile
Option.MajorRoads	Specifies whether to include all roads in the calculation or just major roads. By default, the Get Travel Boundary is calculated with major roads set to true. This improves performance but the accuracy may decrease.

Parameter	Description
	This map represents a travel boundary with travel allowed on all roads:  Skate Creek Adirondack Park Preserve Clift Lost Pond Fine Star Lake Little River Little River Streeter Lake This map represents a travel boundary with travel restricted to major roads only:  Moosehead Pond Skate Creek Adirondack Park Preserve Clift Lost Pond Twin Lakes Star Lake Little River One of the following: **Y** Include only major roads in the calculation. Default. **N** Include all roads in the calculation.
Option.DefaultAmbientSpeed	Specifies the speed to travel when going off a network road to find the travel boundary. Examples of off-network travel include driveways and access roads.

Parameter	Description
	This option is available only when you specify a time value in the Option.DefaultCostUnits parameter or the Data.TravelBoundaryCostUnits parameter. The default is 15. Specify the speed units in the Option.AmbientSpeedUnit parameter. To control how off-network travel is used in the travel boundary calculation, you need to specify the speed of travel off the road network (the ambient speed). Ambient speed can affect the size and shape of the travel boundary polygon. In general, the faster the ambient speed, the larger the polygon. For example, if you were at a point with 5 minutes left, and if the ambient speed were 15 miles per hour, boundary points would be put at a distance of 1.25 miles. If the ambient speed were reduced to 10 miles per hour, boundary points would be put at a distance of 0.83 miles. Note that you can limit the distance allowed off a network road by using the the Option.MaximumOffRoadDistance parameter. **Note:** If you are calculating pedestrian travel boundaries we recommend that you change the default ambient speed to 3 MPH (5 KPH).
Option.AmbientSpeedUnit	The unit of measure to use with the value specified in the Option.DefaultAmbientSpeed parameter. **KPH** Kilometers per hour. **MPH** MILES per hour. Default. **MTPS** Meters per second.
Option.AmbientSpeed.RoadType.<Type>	Specifies the ambient speed to use for off-network travel based on the road type. If you do not specify an ambient speed for a road type, the default ambient speed will be used, as specified in the Option.DefaultAmbientSpeed parameter. The following map shows an example of a travel boundary without ambient speed overrides:

Parameter	Description
	 32 378 State Hwy State Hwy 32 787 787 U S Hwy 4 136 Wyı N Greenbush Rd 4 90 Stream I 90E 90 N Greenbush 43 Mill Creek For comparison, this map shows the same travel boundary with ambient speed overrides:  32 378 State Hwy te Hwy 787 787 787 U S Hwy 4 Whiteview W 136 Wynar N Greenbush Rd 4 90 Stream I 90E 90 N Greenbush 43 Mill Creek <Type> can be one of the following: • AccessWay • Backroad • Connector • Ferry • Footpath • LimitedAccessDenseUrban • LimitedAccessRural • LimitedAccessSuburban

Parameter	Description
	• LimitedAccessUrban • LocalRoadDenseUrban • LocalRoadRural • LocalRoadSuburban • LocalRoadUrban • MajorLocalRoadDenseUrban • MajorLocalRoadRural • MajorLocalRoadSuburban • MajorLocalRoadUrban • MajorRoadDenseUrban • MajorRoadRural • MajorRoadSuburban • MajorRoadUrban • MinorLocalRoadDenseUrban • MinorLocalRoadRural • MinorLocalRoadSuburban • MinorLocalRoadUrban • NormalRoadDenseUrban • NormalRoadRural • NormalRoadRural • NormalRoadUrban • PrimaryHighwayDenseUrban • PrimaryHighwayRural • PrimaryHighwaySuburban • PrimaryHighwayUrban • RampDenseUrban • RampLimitedAccess • RampMajorRoad • RampPrimaryHighway • RampRural • RampSecondaryHighway • RampUrban • RampSuburban • SecondaryHighwayDenseUrban • SecondaryHighwayRural • SecondaryHighwaySuburban • SecondaryHighwayUrban

Response

Get Travel Boundary returns the following fields:

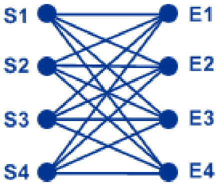
Table 21: GetTravelBoundary Outputs

Response Element	Format	Description
Status	String	Reports the success or failure of the match attempt.

Response Element	Format	Description
		null Success
		F Failure
Status.Code	String	Reason for failure, if there is one. One of the following: • InsufficientInputData (missing lat/lon) • MalformedInputData (wrong input format) • InputOutOfRange (input is out of range) • EngineError (engine-generated error)
Status.Description	String	Description of failure indicated in Status.Code.

GetTravelCostMatrix

GetTravelCostMatrix calculates the travel time and distances between an array of start and end locations. You can use GetTravelCostMatrix to find the shortest or fastest paths between a number of start points and a number of end points, and determine the total time and distance of the individual routes (the route costs). For example if you input four start points (S1 through S4) and four end points (E1 through E4), a total of 16 routes will be returned as illustrated in the following diagram:



Matrix routing is often used to determine service response time and coverages for specific services such as a fire house or police station. You may require these calculations to ensure they can provide adequate coverage for service level agreements such as having one or more people who can respond to an incident within 20 minutes of first notification. The calculation can be used an analysis tool to determine the risk of an insured property or person based on the probability that ambulance, public safety, or fire personnel can reach the property/person in a reasonable amount of time.

Note: Get Travel Cost Matrix is only available as a SOAP web service. Get Travel Cost Matrix is not available through REST. It is also not available through the Java, C++, C, .NET, or COM APIs.

GetTravelCostMatrix is part of the Enterprise Routing Module.

GetTravelDirections

GetTravelDirections returns routing information for a set of two distinct points or for multiple points. It takes a starting latitude and longitude point and an ending latitude and longitude point as input and returns the route that is either fastest or shortest, depending on how you configure the stage.

Each country has its own database, named in this format: Enterprise Routing Module - <Country>. Each database also has its own country code. For example, the name of the Austrian database is "Enterprise Routing Module - Austria," and the Austrian batch country code is "A1T." Each database requires a separate license.

Note: Get Travel Directions is only available as a SOAP web service. Get Travel Directions is not available through REST. It is also not available through the Java, C++, C, .NET, or COM APIs.

GetTravelDirections is part of the Enterprise Routing Module.

Enterprise Tax Module

AssignGeoTAXInfo

AssignGeoTAXInfo identifies the tax districts that apply to a given address. Specifically, AssignGeoTAXInfo returns this information about an address:

- Latitude/longitude coordinates
- FIPS state codes and county codes
- County names
- MCD/CCD codes and names
- CBSA/CSA codes and names
- Place codes and names
- Incorporated or unincorporated status codes
- Cross-reference tax keys
- Result indicators
- Optionally, the relationship of an address to user-defined polygons

AssignGeoTAXInfo optionally includes enhanced tax jurisdiction information for an address, including:

- **Insurance premium districts**—Areas designated for the collection of taxes imposed on insurance policy premiums, based on the policy holder's address. Insurance premium districts are created by state governments.
- **Payroll tax districts**—Areas designated for the collection of taxes imposed on employers to support state or local government facilities and services, based on the employee's and/or employer's address. Examples include taxes collected for districts to pay for schools, police, or other services. Payroll tax districts are created by state or local governments.
- **Payroll system tax codes**—Codes that represent specific jurisdictions that collect payroll tax. Using payroll system tax codes has advantages over using the payroll tax district information returned by Assign GeoTAX Info:

  - AssignGeoTAXInfo uses an additional database to determine payroll tax codes, resulting in more accurate payroll tax determination.
  - Many payroll systems use specific codes to determine withholding amounts. Since you can customize the payroll tax codes returned by AssignGeoTAXInfo, you can set up a process where AssignGeoTAXInfo returns the exact payroll tax codes required by your payroll system, instead of returning jurisdictional IDs that must then be translated into the codes used by your system.

- **Property tax districts**—Areas designated for the collection of taxes imposed on property owners to support local government facilities and services, based on the property's location. Examples include taxes collected for districts to pay for schools, police, or other services. Property tax districts are created by local governments.
- **Special purpose tax districts**—Areas designated for the collection of taxes imposed on residents to support specialized services for residents of the district, based on the resident's address. Examples include services such as sewer service, transit service, or water resources. Special purpose tax districts are created by legislative action, court action, or public referendums. This optional information requires the use of boundary files which require an additional license. Contact your Pitney Bowes Software sales representative for more information.
- **Sales and Use Tax Rates**—Using the optional Pitney Bowes Software Sales and Use Tax Rate file, AssignGeoTAXInfo can return sales and use tax rates for each of the assigned tax jurisdictions as well as the total tax rate for the assigned locations.

AssignGeoTAXInfo is part of the Enterprise Tax Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/AssignGeoTaxInfo/results.json
```

XML endpoint:

```
http://server:port/rest/AssignGeoTaxInfo/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/AssignGeoTAXInfo/results.json?
Data.AddressLine1=1+Global+View&Data.City=Troy&
Data.StateProvince=NY&Data.PostalCode=12180
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Confidence": "100.0",
   "ProcessedBy": "GTX",
   "Census.MatchCode": "S",
   "Census.MatchLevel": "Street",
   "County.Code": "083",
   "County.Name": "Rensselaer",
   "StateCode": "36",
   "LatLong": "42.683028-073.702968",
   "LatLong.MatchCode": "R",
   "LatLong.MatchLevel": "Rooftop",
   "Latitude": "42.683028",
   "Longitude": "-073.702969",
   "State.Abbreviation": "NY",
   "Place.Code": "00000",
   "Place.IncorporatedFlag": "Uninc",
   "AddressLine1": "1 GLOBAL VW",
   "City": "TROY",
   "StateProvince": "NY",
   "PostalCode": "121808371",
   "AddressMatch.MatchCode": "S80",
   "AddressMatch.LocationCode": "AS0",
   "AddressMatch.LastLine": "TROY,NY 12180-8371"
   "AddressMatch.Zip": "12180",
   "AddressMatch.Zip4": "8371",
   "AddressMatch.GenRC": "S"
   "AddressMatch.DataTypeName": "TOMTOM",
   "MCD.DistanceToBorder": "000002938",
   "Place.DistanceToBorder": "00000000",
   "GNISCode": "000000000",
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/AssignGeoTAXInfo/results.xml?
Data.AddressLine1=1+Global+View&Data.City=Troy&
Data.StateProvince=NY&Data.PostalCode=12180
```

The XML returned by this request would be:

```
<ns2:xml.AssignGeoTAXInfoResponse
xmlns:ns2="http://www.pb.com/spectrum/services/AssignGeoTAXInfo">
   <ns2:output_port>
      <ns2:Address>
```

```
         <ns2:Confidence>100.0</ns2:Confidence>
         <ns2:ProcessedBy>GTX</ns2:ProcessedBy>
         <ns2:Census.MatchCode>S</ns2:Census.MatchCode>
         <ns2:Census.MatchLevel>Street</ns2:Census.MatchLevel>
         <ns2:County.Code>083</ns2:County.Code>
         <ns2:County.Name>Rensselaer</ns2:County.Name>
         <ns2:StateCode>36</ns2:StateCode>
         <ns2:LatLong>42.683028-073.702968</ns2:LatLong>
         <ns2:LatLong.MatchCode>R</ns2:LatLong.MatchCode>
         <ns2:LatLong.MatchLevel>Rooftop</ns2:LatLong.MatchLevel>
         <ns2:Latitude>42.683028</ns2:Latitude>
         <ns2:Longitude>-073.702969</ns2:Longitude>
         <ns2:State.Abbreviation>NY</ns2:State.Abbreviation>
         <ns2:Place.Code>00000</ns2:Place.Code>
         <ns2:Place.IncorporatedFlag>Uninc</ns2:Place.IncorporatedFlag>
         <ns2:AddressLine1>1 GLOBAL VW</ns2:AddressLine1>
         <ns2:City>TROY</ns2:City>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>121808371</ns2:PostalCode>
         <ns2:AddressMatch.MatchCode>S80</ns2:AddressMatch.MatchCode>
        <ns2:AddressMatch.LocationCode>AS0</ns2:AddressMatch.LocationCode>

         <ns2:AddressMatch.LastLine>TROY, NY
12180-8371</ns2:AddressMatch.LastLine>
         <ns2:AddressMatch.Zip>12180</ns2:AddressMatch.Zip>
         <ns2:AddressMatch.Zip4>8371</ns2:AddressMatch.Zip4>
         <ns2:AddressMatch.GenRC>S</ns2:AddressMatch.GenRC>

<ns2:AddressMatch.DataTypeName>TOMTOM</ns2:AddressMatch.DataTypeName>
         <ns2:MCD.DistanceToBorder>000002938</ns2:MCD.DistanceToBorder>
        <ns2:Place.DistanceToBorder>00000000</ns2:Place.DistanceToBorder>

         <ns2:GNISCode>000000000</ns2:GNISCode>
         <ns2:LatLong.StreetMatchCode/>
         <ns2:LatLong.StreetMatchLevel/>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.AssignGeoTAXInfoResponse>
```

Request

Parameters for Input Data

The following table provides information on the format of AssignGeoTAXInfo input.

Table 22: AssignGeoTAXInfo Input Data

Parameter	Format	Description
Data.AddressLine1	String [100]	First address line
Data.AddressLine2	String [100]	Second address line
Data.AddressLine2	String [100]	Third address line
Data.AddressLine4	String [100]	Fourth address line
Data.BufferWidth	String [10]	Specifies the width of the polygon buffers to use for Boundary File processing. The buffer width is used to determine if a point is close to the edge of a polygon. The output field

Parameter	Format	Description
		BufferRelation indicates whether or not the point is within the polygon's buffer area. For more information, see **Buffering** on page 290. This field overrides the value specified in the Option.DefaultBufferWidth parameter. Specify the border width in the units specified by the Option.DistanceUnits parameter. If you do not specify a buffer width in this input field, the default is used.
Data.CandidateID	String [1]	For a multi-match outcome, indicates which match result should be returned; valid values are 1-9.
Data.City	String [50]	City name
Data.Country	String [var]	The country where the address resides. The data you enter in this field has no impact on processing. It is simply passed through to output. **Note:** AssignGeoTAXInfo only supports US addresses.
Data.FirmName	String [var]	Company or firm name
Data.PostalCode	String [9]	Nine-digit ZIP Code
Data.StateProvince	String [50]	The state where the address resides. The data you enter in this field has no impact on processing. It is simply passed through to output.
Data.UseBufferWidth	Long [10]	Specifies the width of the polygon buffers to use for User-Defined Boundary File processing. The buffer width is used to determine if a point is close to the edge of a polygon. The output field BufferRelation indicates whether or not the point is within the polygon's buffer area. For more information, see **Buffering** on page 290. This field overrides the value specified in the Option.DefaultBufferWidth parameter. Specify the border width in the units specified by the Option.DistanceUnits parameter. If you do not specify a buffer width in this input field, the default is used.

Matching Options

Matching options control the address search methodology and match results handling returned by AssignGeoTAXInfo. **Table 23: AssignGeoTAXInfo Matching Options** on page 74 lists the address search and matching options.

Table 23: AssignGeoTAXInfo Matching Options

Parameter	Description
Optional files: The following options enable the database resource(s) to use in the search process.	
Option.UseGeoTaxAuxiliaryFile	Specifies whether or not AssignGeoTAXInfo should attempt a match to the GeoTAX Auxiliary file. The GeoTAX Auxiliary file contains new addresses that have not yet been added to the Master File. **Y** Use the GeoTAX Auxiliary file for matching. **N** Do not use the GeoTAX Auxiliary file for matching. (default)
Option.UseAuxiliaryFile	Specifies whether or not AssignGeoTAXInfo should attempt a match to a User Auxiliary file. User Auxiliary files are user-defined files that the Enterprise Tax Module uses to override results from the master files in street-level matching. **Y** Use the User Auxiliary file for matching. **N** Do not use the User Auxiliary file for matching. (default)
Option.UseStateProvidedFile	Specifies whether or not AssignGeoTAXInfo should attempt a match to the state-supplied file. Use this option in combination with FileSearchOrder to specify a state-supplied file to use. State-supplied files are provided by individual state governments. By matching to the state-supplied files, you can remain compliant with tax jurisdiction assignment requirements mandated by new federal and state laws, such as the Mobile Telecommunications Sourcing Act and the Florida state Communications Services Tax Simplification Law. There are two supported file formats: the Florida-native format and the national TS-158 format (ANSI Transaction Set No. 158). The state of Florida provides address files in both the TS-158 and its own native format. The state of Washington provides address data in the TS-158 format. Assign GeoTAX Info attempts to match to the state supplied file first. If a state match cannot be found, it attempts a match to the master files. **Y** Use the State-supplied file for matching. **N** Do not use the State-supplied file for matching. (default) **Note:** You must install the appropriate State-supplied file to use these options. For instructions, see the *Spectrum™ Technology Platform Installation Guide*.
Option.FileSearchOrder	Specifies which state-supplied file to use. This option only takes effect if you specify Option.UseStateProvidedFile=Y. One of the following: **FLOnly** Use only the Florida-native formatted file. **TSOnly** Use only the TS-158 formatted file.
Option.UseRelaxedSecondaryMatching	Specifies whether or not AssignGeoTAXInfo matches input addresses with secondary information to records without secondary information. This option applies only to Florida-native files. **Y** Use relaxed secondary matching.

Parameter	Description
	N Do not use relaxed secondary matching. (default)
Address Searching and Matching Options: The following options can be enabled for use in the address search and match processes.	
Option.GsMatchMode	Match modes determine the leniency used to make a match between your input and the reference database. Select a match mode based on the quality of your input and your desired output. For example, if you have an input database that is prone to errors, you may want to select the relaxed match mode. **0 - Exact** Requires a very tight match. This restrictive mode generates the fewest match candidates, which decreases the processing time. When using this mode, ensure that your input is very clean; free of misspellings and incomplete addresses. **1 - Close** Requires a close match and generates a moderate number of match candidates. (default) **2 - Relaxed** Allows a loose match and generates the most match candidates, which increases the processing time and results in more multiple matches. Use this mode if you are not confident that your input is clean; free of misspellings and incomplete addresses. This is the only mode that does not respect the street parity when making an address match.
Option.GsSearchArea	The search area options allow for searching the address' finance area or an expanded area specified by distance. **1** Searches the entire finance area for a match. A finance area is a region defined by the U.S. Postal Service and typically consists of a set of contiguous ZIP Codes.(default) **2** Searches the area specified by the radius in miles. The search area can be extended up to a 99-mile radius from the centroid of the input ZIP Code to assist in finding a match when the input address contains limited or inaccurate city or ZIP Code information. The expanded area is confined to within the state's borders.
Option.GsSearchRadius	Radius for search area. **1-99 miles** Search radius. (default = 25 miles)
Option.GsEnableFirstLetterExpanded	Looks for the correct first letter of a street address if the first letter is missing or incorrect. The Enterprise Tax Module searches through the alphabet looking for possible correct first letters to complete the street address. **Note:** This feature is disabled by default and cannot be enabled in Exact mode. **Y** Enable first letter change matches. **N** Do not allow first letter change matches. (default)

Parameter	Description
Option.GsEnableRangedAddress	Matches to a house range input. Some business locations are identified by address ranges. For example, a shopping plaza could be addressed as 10-12 Front St. - this is how business mail is typically addressed to such a business location. When this feature is enabled, the address range is geocoded to the interpolated mid-point of the range. **Y** Allow address range matches. **N** Do not allow address range matches. (default)
Option.GsAlternateLookup	This option allows specifying the preferred way to match when both an address and firm name are provided. The matching method can be set to match to the address rather than the firm name or vice versa. If neither are specified, the default matching method is to match to the address line only. **1** Searches for street name, but if there isn't a match, will use the firm name. **2** Looks up the firm name, but if there isn't a match, will use the street name. **3** Searches only street records. (default)
Option.GsMultiMatchResolution	A multi-match occurs when multiple equally-scored matches are found in either the Points or Streets files and cannot be resolved to a single best candidate. There are several choices for handling a multi-match outcome: **N** No matches are returned. (default) **R** Return the first match candidate in the list. **A** The information for all the match candidates is returned.

Specifying Default State-Supplied File Options

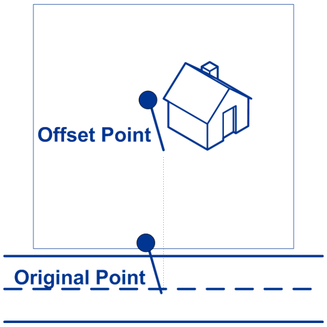
If you use the Spectrum™ Technology Platform API, note that the value you specify in the Management Console for the **State supplied file** field controls the default settings for three AssignGeoTAXInfo API options: UseStateProvidedFile, UseRelaxedSecondary, and FileSearchOrder. The following table show how each value in the **State supplied file** field affects these three options.

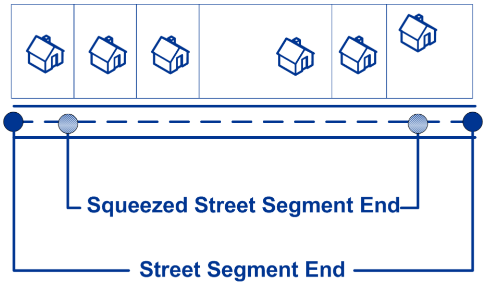
"State supplied file" value	UseStateProvidedFile	FileSearchOrder	UseRelaxedSecondary
None	N	N/A	N
Florida-native	Y	FSOnly	N
Florida-native with relaxed secondary matching	Y	FSOnly	Y
TS-158	Y	TSOnly	N

Geocoding Options

Geocoding is the process of determining the latitude/longitude coordinates of a given address. Address coordinates are used as the basis for determining the tax jurisdictions for an address. Geocoding options control how AssignGeoTAXInfo determines address latitude/longitude coordinates.

Table 24: AssignGeoTAXInfo Geocoding Options

Parameter	Description
Option.Database.GTX	Select the database resource to use in the geocoding process.
Latitude/Longitude placement: The following options can be set for the geocode result.	
Option.LatLongOffset	Indicates the offset distance in feet from the street center line. The offset distance is used in street-level geocoding to prevent the geocode from being in the middle of a street. It compensates for the fact that street-level geocoding returns a latitude and longitude point in the center of the street where the address is located. Since the building represented by an address is not on the street itself, you do not want the geocode for an address to be a point on the street. Instead, you want the geocode to represent the location of the building which sits next to the street. For example, an offset of 50 feet means that the geocode will represent a point 50 feet back from the center of the street. The distance is calculated perpendicular to the portion of the street segment for the address. Offset is also used to prevent addresses across the street from each other from being given the same point. The following diagram shows an offset point in relation to the original point.  **0** No offset. (default) **20** Twenty feet offset from street center line. **40** Forty feet offset from street center line. **60** Sixty feet offset from street center line.
Option.Squeeze	Specifies if AssignGeoTAXInfo should squeeze the street end points when determining the geocode of an address in street-level matching. The squeeze value is 5% (2.5% from each endpoint). The following diagram compares the end points of a street segment to the squeezed end points of a street segment.

Parameter	Description
	 **Y** Apply squeeze. **N** Do not apply squeeze. (default)
Option.LatLongFormat	Indicates the desired format for returned latitude/longitude. Options include: **PreZero** Latitude/longitude in the following format: 090000000N180000000W. (default) **PreZeroDecimal** Latitude/longitude in the following format: 090.000000N180.000000W. **Decimal** Latitude/longitude in the following format: 90.000000-180.000000. **DecimalAssumed** Latitude/longitude in the following format: 90000000-180000000. **DegMinSec** Latitude/longitude in the following format: 90 00 00N180 00 00W.
Expanded Geocoding options: The following options enable additional geocoding functionality.	
Option.GsEnableAddressPointInterpolation	Address point interpolation uses a patented process that improves upon regular street segment interpolation by inserting point data into the interpolation process. **Note:** This feature is only for use with point-level geocoding. A match is first attempted using the loaded points data. If an exact point match is found in the points data, then searching ceases and the point match is returned. If an exact point match was not found, the Enterprise Tax Module attempts to find high and low boundary address points to use for address point interpolation. **Y** Enable address point interpolation. **N** Disable address point interpolation. (default)
Minimum geocode quality:	
Option.GsEnableGeographicFallback	The default search does not perform a search of geographic centroids. When enabled, the Geographic Fallback feature locates the first city, county and/or state centroid, and then matches from the set of possible matches found. **Y** If a definitive match cannot be made, then return the next higher level geographic centroid. **N** Disable geographic fallback feature. (default)

Parameter	Description
Option.GsEnableStreetCentroid	If an input street address cannot be found using the street number and name, the Enterprise Tax Module then searches the input ZIP Code or city/state for the closest match. If the Enterprise Tax Module is able to locate the street, it returns a geocode along the matched street segment rather than the geocode for the entered ZIP Code or ZIP + 4. When using street locator geocoding, if no exact matching house number is found, a match code of either E029 (no matching range, single street segment found), or E030 (no matching range, multiple street segment) returns. For example, if you enter Main St and there are both an E Main St and a W Main St within the input ZIP Code, then an E030 returns and the location code returned is reflective of the input ZIP Code. The location code returned begins with a 'C' when matched to a single street segment, indicated by E029. The Enterprise Tax Module does not change the street name on the output address. **Y** If a street or point match cannot be made, then return a street level centroid. **N** Do not return a street level centroid if a match cannot be made. (default)
Boundary matching: The following options can be set when matching to a boundary file such as SPD, IPD, PAY, Place and MCD or user-defined.	
Option.DistanceUnits	Specifies the units in which to measure distance. One of the following: **Miles** Distances are measured in miles. **Km** Distances are measured in kilometers. **Feet** Distances are measured in feet. (default) **Meters** Distances are measured in meters.
Option.DefaultBufferWidth	Specifies the buffer width to use for tax district boundary files. The tax district boundary files are the Special Purpose District (SPD) file, the Insurance Premium District (IPD) file, the Payroll Tax District (PAY) file, the Personal Property Tax District (PTD) file and Place and MCD files. The default buffer width that you specify here can be overridden on a record-by-record basis using the BufferWidth input field. For more information on buffers, see **Buffering** on page 290.
Option.DefaultUserBufferWidth	Specifies the buffer width to use for user-defined boundary files. Specify the distance in the units of measurement specified in the **Distance units** option. For information on buffers, see **Buffering** on page 290. The default buffer width that you specify here can be overridden on a record-by-record basis using the BufferWidth input field. **Note:** To use buffers, the user-defined boundary file must support buffers.

Output Data Options

Data options control the data returned by AssignGeoTAXInfo. **Table 25: AssignGeoTAXInfo Data Options** on page 80 lists the output data options.

Table 25: AssignGeoTAXInfo Data Options

Parameter	Description
Option.GeoTAXOutputRecordType	Select one or more of the following to obtain the type of data you want returned. AssignGeoTAXInfo groups the output fields into record types. If you do not want all of the fields in a record type returned, do not select the check box, and list only those fields you want returned in Extra Output Fields. • **C**—Census • **L**—Latitude/Longitude • **T**—Tax Jurisdiction • **U**—User-defined boundary file • **W**—Payroll System Tax Codes • **X**—Auxiliary File • **B**—PB Software Sales and Use Tax Rate file You can also specify one, and only one, of the following: **I** Insurance Premium Tax District (IPD) **P** Property Tax District (PTD) **R** Payroll Tax District (PAY) **S** Special Purpose Tax District (SPD) For a description of the fields in each output group, see **Response** on page 81. **Note:** If you specify `W`, to obtain the best payroll system tax code match possible.
Option.TaxKey	If you integrate AssignGeoTAXInfo with third-party tax compliance software from Vertex or Taxware, select which vendor you use. This controls the value returned in the GeoTAXKey output field. One of the following: **T** Return the Taxware jurisdiction code for the address. **Y** Return the Vertex jurisdiction code for the address. Select this option if you obtained a MatchMaster file from Vertex. **V** Return the Vertex jurisdiction code for the address. Select this option if you obtained a Vertex file from Pitney Bowes Software. **N** Do not return either the Taxware or Vertex jurisdiction codes (default).
Option.TaxRate	Indicates the sales and use tax rate type to return or None: **N** Do not return sales and use tax rates. (default) **G** Return the General sales and use tax rates.

Parameter	Description
	A Return the Automotive sales and use tax rates. **C** Return the Construction sales and use tax rates. Return the Medical sales and use tax rates.
Option.OutputFields	Indicates the individual output fields you want returned. You can use this field instead of the Output Record Type to limit the output to those fields that are important to your current data needs. For a list of the fields included in each data type, see **Response** on page 81.

Output Format

Output format options control how AssignGeoTAXInfo formats output data. The following table lists the output format options.

Table 26: AssignGeoTAXInfo Output Format Options

Parameter	Description
Option.OutputCasing	Specifies the casing of these output fields: County.Name, CBSA.Name, MCD.Name, Place.Name, IPDn.DistrictName, PAYn.DistrictName, SPDn.DistrictName, and PTCn.PayrollDescription. One of the following: **M** The output in mixed case (default). For example: Rensselaer. **U** The output in upper case. For example: RENSSELAER.

Response

Result Indicators

The following table lists the results returned from the address matching and geocoding process.

Table 27: Address Matching and Geocoding Status Output Fields

Response Element	Max. Field Length (bytes)	Description
Confidence	4	Indicates the confidence in the output provided; from 0 to 100. The higher the score, the higher the confidence in the match. Calculated based on the match results for individual output fields, using the following algorithm: Census.MatchCode + LatLong.StreetMatchCode + LatLong.MatchCode The maximum confidence score is 100, so if this calculation results in a value greater than 100, the Confidence score is returned as 100.

Response Element	Max. Field Length (bytes)	Description
		AssignGeoTAXInfo uses the following values: • Census.MatchCode • A = 85 • G = 85 • S = 85 • U = 85 • 9 = 65 • 5 = 45 • null = 0
Confidence.SurfaceType	3	Indicates the confidence surface type. Setting a non-zero buffer width enables confidence generation. To determine a confidence level, a confidence surface is first generated. The confidence surface provides the smallest possible area wherein an address is likely to be located. **0** Undefined **1** The search failed - the address was not found. **2** Intersection confidence surface generated. **3** Interpolated street segment. **4** Point-level match. **5** State confidence surface generated. **6** County confidence surface generated. **7** City confidence surface generated. **8** Reserved **9** A ZIP Code confidence surface generated. **10** A ZIP+2 confidence surface generated. **11** A ZIP+4 confidence surface generated. **12** Reserved **13** A street centroid confidence surface generated.
AddressMatch.MatchCode	5	The Match Code indicates the portions of the address that matched or did not match to the reference file. For the list of match codes, see **Match Codes.**
AddressMatch.GenRC	2	General Return Code indicating the type of match. **5** ZIP Code match **9** ZIP+4 Code match **A** User Auxiliary file match

Response Element	Max. Field Length (bytes)	Description
		C Street Centroid match **F** Geographic Fallback match **G** State-supplied file match **I** Intersection match **L** Landmark Auxiliary file match **M** Multiple match (multi-match) **O** Input Latitude/Longitude coordinates match **P** Address point match **S** Street address match **X** Aborted processing or expired database **Blank** Did not match
AddressMatch.DataTypeName	20	Indicates the file from which the match was obtained. One of the following: • USPS • TIGER • TOMTOM - Streets • NAVTEQ - Streets • TOMTOM_POINT • CENTRUS_POINT • NAVTEQ_POINT • STATE_FILE • USER_AUXILIARY • LANDMARK_AUXILIARY
AddressMatch.NumCandidates	2	When there are multiple equally-scored matches, returns the number of multiple match candidates found.
AddressMatch.LocationCode	5	The Location Code indicates the methodology used to complete the geocode and may also provide some information about the quality of the geocode. For the list of location codes, see **Location Codes.**
LatLong.MatchCode	2	Latitude/Longitude General Return Code. Denotes the level for which the geocode was determined. **2** ZIP + 2 centroid **4** ZIP + 4 Code centroid **B** Block group centroid **C** City centroid **I** Intersection **L** Match using the Landmark Auxiliary file **O** Latitude/longitude was input

Response Element	Max. Field Length (bytes)	Description
		R Address-level based on street address **S** State centroid **T** Census tract centroid **U** Address-level match using the GeoTAX Auxiliary Database **Z** ZIP Code centroid based on a five-digit ZIP code **null** No latitude/longitude determined If the **General Return Code** (see **Result Indicators**) is **"P"** (point match), then the following are possible values and have the following meaning: **0** Latitude/Longitude coordinates from User Dictionary. **2** Latitude/Longitude coordinates from Parcel Centroid. **4** Latitude/Longitude coordinates from Address Point. **5** Latitude/Longitude coordinates from Structure Centroid. **7** Latitude/Longitude coordinates from manually-placed Point. **8** Latitude/Longitude coordinates from Front Door Point. **9** Latitude/Longitude coordinates from Driveway Offset Point. **A** Latitude/Longitude coordinates from Street Access Point. **B** Latitude/Longitude coordinates from Base Parcel Point.
LatLong.StreetMatchCode	2	Output street address return code. **H** House number not found on street **L** Latitude/longitude not determined on auxiliary match **S** Street not found in ZIP Code **Z** ZIP Code not found in street address database **N** Street-level matching option not selected **null** The street was successfully matched

Auxiliary File

The following table lists the output fields that contain auxiliary file data. To include auxiliary file data in the output, set **Option.GeoTAXOutputRecordType** = X. The following table lists the output fields that contain tax jurisdiction data.

Table 28: Auxiliary File Output Fields

Response Element	Max. Field Length (bytes)	Description
AuxiliaryData.AuxiliaryFile	301	Data retrieved as a result of an auxiliary match from the user-defined area of the auxiliary file.
AuxiliaryData.StateFile	201	Data retrieved as a result of a state match. Data content and format vary depending on the state file used.

Census

The census output fields contains census information from the U.S. Census, including Minor Civil Divisions (MCDs) and Census County Division (CCD) names and codes. MCDs are the primary political or administrative divisions of a county, representing many kinds of legal entities with a variety of governmental and administrative functions. CCDs are established in states where there are no legally established MCDs. The Census Bureau recognizes MCDs in 28 states and has established CCDs in 21 states. The District of Columbia has no primary divisions, and the city of Washington, DC is considered equivalent to an MCD for data presentation purposes.

Census data also contains the Federal Information Processing Standards (FIPS) codes for each state and county. The FIPS State Code and the FIPS County Code are both used by the Census Bureau to identify these geographic units.

The following table lists the output fields that contain census data. To include census data in the output, set **Option.GeoTAXOutputRecordType** = C.

Table 29: Census Data Output Fields

Response Element	Max. Field Length (bytes)	Description
Census.Block	4	Census Block ID.
Census.BlockGroup	2	Census Block Group code.
Census.MatchCode	2	The level of match obtained against the databases. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output. **A** Auxiliary street match **G** State file address match **S** Street address match **U** GeoTAX Auxiliary file match

Response Element	Max. Field Length (bytes)	Description
		9 ZIP + 4 Code level match **5** ZIP Code level match **null** Unsuccessful match
Census.MatchLevel	19	The level of match obtained against the databases. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output. **Auxiliary** Auxiliary street match **Aux2** GeoTAX Auxiliary file match **Street** Street address match **Gov** State file address match **ZIP+4** ZIP + 4 Code level match **ZIP** ZIP Code level match **null** No match
Census.Tract	7	Six-digit tract number extracted from the Census.BlockCode.
County.Code	4	Three-digit Federal Information Processing Standards (FIPS) county code extracted from the Census.BlockCode. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output.
County.Name	26	Name of the county. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output.
MCD.Code	6	Minor Civil Division/Census County Division (MCD/CCD) Code.
MCD.Name	41	Minor Civil Division/Census County Division (MCD/CCD) name.
MCD.PointStatus	2	An address can be compared to a Minor Civil Division (MCD)/county subdivision file (Cousub.txt). This output field returns the result for a comparison between the matched address point to the polygon defined by the Cousub.txb file. For more information on buffers, see **Buffering** on page 290.

Response Element	Max. Field Length (bytes)	Description
		Note: The buffer distance for Cousub.txb is internally set to zero and cannot be modified. **P** The point is in the polygon. **I** The point is in the buffer area inside the polygon. **B** The point is in the buffer area and outside of the polygon. **blank** Polygon not found.
MCD.DistanceToBorder	10	Returns the distance in feet between the matched address point to the polygon defined by the Cousub.txb file.
MCD.Confidence	4	Provides the percentage overlap of the geocode to the MCD polygon layer. The returned percentage value describes the probability that the point falls in the MCD.
CBSA.Code	6	Core Based Statistical Area (CBSA) code.
CBSA.Name	76	Core Based Statistical Area (CBSA) name.
CBSA.MetroFlag	2	Indicates if the CBSA is a "Metropolitan Statistical Area" or a "Micropolitan Statistical Area". **Y** Metropolitan Statistical Area - A Core Based Statistical Area associated with at least one urbanized area that has a population of at least 50,000. The Metropolitan Statistical Area comprises the central county or counties containing the core, plus adjacent outlying counties having a high degree of social and economic integration with the central county as measured through commuting. **N** Micropolitan Statistical Area - A Core Based Statistical Area associated with at least one urban cluster that has a population of at least 10,000, but less than 50,000. The Micropolitan Statistical Area comprises the central county or counties containing the core, plus adjacent outlying counties having a high degree of social and economic integration with the central county as measured through commuting.
CBSAD.Code	6	Core Based Statistical Area Division (CBSAD) code.
CBSAD.Name	73	Core Based Statistical Area Division (CBSAD) name.
CSA.Code	4	Combined Statistical Area (CSA) code.
CSA.Name	78	Combined Statistical Area (CSA) name.
State.Abbreviation	3	Two-character state abbreviation.

Response Element	Max. Field Length (bytes)	Description
		Note: This field is always included in the output regardless of whether or not you choose to include census data in the output
StateCode	3	Two-digit Federal Information Processing Standards (FIPS) state code extracted from the Census.BlockCode. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output.

Latitude/Longitude

The following table lists the output fields that contain latitude and longitude data. Latitude/Longitude data contains the coordinates for the address and additional information about how AssignGeoTAXInfo determined the latitude and longitude. To include latitude/longitude data in the output, set **Option.GeoTAXOutputRecordType** = L.

Table 30: Latitude/Longitude Output Fields

Response Element	Max. Field Length (bytes)	Description
Latitude	8	Seven-digit number in degrees and calculated to four decimal places (in the format you specified).
Latitude.Directional	2	Latitude directional. **N** North **S** South
LatLong	23	Returned latitude/longitude, in the format you specified (up to 22 alphanumeric characters).
LatLong.MatchCode	2	Latitude/Longitude General Return Code. Denotes the level for which the geocode was determined. **2** ZIP + 2 centroid **4** ZIP + 4 Code centroid **B** Block group centroid **C** City centroid **I** Intersection **L** Match using the Landmark Auxiliary file **O** Latitude/longitude was input **R** Address-level based on street address **S** State centroid

Response Element	Max. Field Length (bytes)	Description
		T Census tract centroid
		U Address-level match using the GeoTAX Auxiliary Database
		Z ZIP Code centroid based on a five-digit ZIP code
		null No latitude/longitude determined
		If the **General Return Code** (see **Result Indicators**) is "**P**" (point match), then the following are possible values and have the following meaning:
		0 Latitude/Longitude coordinates from User Dictionary.
		2 Latitude/Longitude coordinates from Parcel Centroid.
		4 Latitude/Longitude coordinates from Address Point.
		5 Latitude/Longitude coordinates from Structure Centroid.
		7 Latitude/Longitude coordinates from manually-placed Point.
		8 Latitude/Longitude coordinates from Front Door Point.
		9 Latitude/Longitude coordinates from Driveway Offset Point.
		A Latitude/Longitude coordinates from Street Access Point.
		B Latitude/Longitude coordinates from Base Parcel Point.
LatLong.MatchLevel	14	A description of the value returned in the LatLong.MatchCode field.
		ZIP+2 ZIP + 2 centroid
		ZIP+4 ZIP + 4 centroid
		Block Block group centroid
		CityCentroid City centroid
		Intersection Intersection match
		LandmarkAux Match using the Landmark Auxiliary file
		LatLonInput Input Latitude/Longitude coordinates was used
		Rooftop Exact address match
		StateCentroid State centroid

Response Element	Max. Field Length (bytes)	Description
		Tract Census tract centroid **Auxiliary** Address-level match using the GeoTAX Auxiliary Database **ZIP** ZIP Code centroid
LatLong.StreetMatchCode	2	Output street address return code. **H** House number not found on street **L** Latitude/longitude not determined on auxiliary match **S** Street not found in ZIP Code **Z** ZIP Code not found in street address database **N** Street-level matching option not selected **null** The street was successfully matched
LatLong.StreetMatchLevel		Street level match used to determine the latitude/longitude **FullMatch** Successful match **HouseNotFound** House number not found on street **LatLongNotFound** Latitude/longitude not determined on auxiliary match **StreetNotFound** Street not found in ZIP Code **ZipNotFound** ZIP Code not found in street address database **NotUsed** Street-level matching option not selected
Longitude	8	Seven-digit number in degrees and calculated to four decimal places (in the format specified).
Longitude.Directional	2	Longitude directional. **E** East **W** West

Input Address

AssignGeoTAXInfo always returns the input address as part of the output. The input address fields are returned as input from the data. AssignGeoTAXInfo does not change these input values.

Table 31: Input Address Output Fields

Response Element	Max. Field Length (bytes)	Description
AddressLine1	101	Input address line 1.
AddressLine2	101	Input address line 2.
AddressLine3	101	Input address line 3.
AddressLine4	101	Input address line 4.
City	51	Input address city.
Country	25	Input address country.
FirmName	101	Input address firm name.
PostalCode	10	Input address postal code
StateProvince	51	Input address state.

Payroll System Tax Code

The following table lists the output fields that contain Payroll System Tax Code (PTC) data. For more information on payroll tax districts, see **AssignGeoTAXInfo** on page 70. To include this data in the output, set **Option.GeoTAXOutputRecordType** = W.

Note: AssignGeoTAXInfo returns up to six payroll tax codes per address.

Table 32: Payroll System Tax Code Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberPTCsFound	2	The number of payroll tax codes found for this address.
PTC*n*.MatchCode	2 per PTC	Indicates the level of match obtained for the address. In order from most specific match to least, the possible match codes are: **P** The address was matched to a specific Payroll District ID. This is the most specific match. **G** The address was matched to a GNIS Code. **F** The address was matched to a county's FIPS code. **S** The address was matched to a state's FIPS code. This is the least specific match.
PTC*n*.PayrollCode	16 per PTC	A code that represents a taxing authority in a payroll application. This is a user-defined code. The specific codes are determined by the payroll application that utilizes the data returned by AssignGeoTAXInfo.

Response Element	Max. Field Length (bytes)	Description
PTC*n*.PayrollDescription	41 per PTC	A description of the purpose of this payroll code.
PTC*n*.PayrollFlag	7 per PTC	A user-defined flag from the PTC database.
StateCounty	33	The state abbreviation and county name.

Tax Jurisdiction

Tax jurisdiction data contains information about the "place" for the location. A "place" is a geographic area defined on the basis of population criteria that vary by state; or, an area recognized as significant because it is located in an incorporated municipality. Places are used to determine tax jurisdiction.

The following table lists the output fields that contain tax jurisdiction data. To include tax jurisdiction data in the output, set **Option.GeoTAXOutputRecordType** = T.

Table 33: Tax Jurisdiction Output Fields

Response Element	Max. Field Length (bytes)	Description
GeoTAXKey	10	The value in this field varies depending on the option you specified in the Option.TaxKey option: If you specified `T`, GeoTAXKey contains the proprietary codes used in Taxware tax compliance software. You can use this code in your Taxware application to find out the tax rate for the jurisdiction. If you specified `Y` or `V`, GeoTAXKey contains the proprietary Vertex® jurisdiction code (comprised of a two-digit Vertex® state code, three-digit FIPS county code, and four-digit Vertex® city code). You can use this code in your Vertex® application to find out the tax rate for the jurisdiction.
GeoTAXKey.MatchCode	2	Return code denoting the level of match obtained against the Vertex or Taxware cross reference files. **E** Exact match using five fields: FIPS state code, FIPS county code, FIPS or GNIS place code, ZIP Code, and FIPS place name. **P** Partial match using four fields: FIPS state code, FIPS county code, FIPS or GNIS place code, and ZIP Code. **A** Alternate match using two fields: ZIP Code, FIPS place name. **N** Record is default coded based on valid state code.

Response Element	Max. Field Length (bytes)	Description
		null No matching record found.
GeoTAXKey.MatchLevel	12	A description of the value returned in the GeoTAXKey.MatchCode field. **Exact** Exact match. See description in GeoTAXKey.MatchCode. **Partial** Partial match. See description in GeoTAXKey.MatchCode. **Alternate** Alternate match. See description in GeoTAXKey.MatchCode. **DefaultCode** Record is default coded. See description in GeoTAXKey.MatchCode. **NoMatch** No matching record found.
GNISCode	10	Unique nine-digit Geographic Names Information System (GNIS) code. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output.
Place.ClassCode	3	Place class code. Place class codes are used to determine the proper taxing jurisdictions **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.Code	6	An identifier for a specific place. A "place" is a geographic area defined on the basis of population criteria that vary by state. Or, an area recognized as significant because it is located in an incorporated municipality. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.IncorporatedFlag	8	Indicates whether the address is located in an incorporated or unincorporated place. A "place" is a geographic area defined on the basis of population criteria that vary by state. Or, an area recognized as significant because it is located in an incorporated municipality. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output. **Inc** Incorporated place code.

Response Element	Max. Field Length (bytes)	Description
		Uninc Unincorporated place code. **Unknown** Incorporation status unknown.
Place.LastAnnexedDate	8	Last annexed date, in the format MM/YYYY, representing the month and year of the most recent boundary change or the most recent available boundary information. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.LastUpdatedDate	8	Last updated date, in the format MM/YYYY, reflecting the month and year when TomTom updated the database to reflect attribute (name change, FIPS change, etc.) or boundary edits to the Place. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.LastVerifiedDate	8	Last verified date, in the format MM/YYYY, representing the month and year that TomTom verified municipality change information. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.Name	41	The name of the "place" where the address is located. A "place" is a geographic area defined on the basis of population criteria that vary by state. Or, an area recognized as significant because it is located in an incorporated municipality. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.PointStatus	2	Returns the result for a comparison between the matched address point to the polygon defined by the Place.txb file. For more information on buffers, see **Buffering** on page 290. **Note:** The buffer distance for Place.txb is internally set to zero and cannot be modified. **P** The point is in the polygon. **I** The point is in the buffer area inside the polygon. **B** The point is in the buffer area and outside of the polygon.

Response Element	Max. Field Length (bytes)	Description
		blank Polygon not found. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.DistanceToBorder	10	Returns the distance between the matched address point to the polygon defined by the Place.txb file. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.Confidence	4	Provides the percentage overlap of the geocode to the Place polygon layer. The returned percentage value describes the probability that the point falls in the specified Place. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.

User-Defined Boundary File

The following table lists the output fields that contain data returned from user-defined boundary files. To include this data in the output, set **Option.GeoTAXOutputRecordType** = U.

Note: AssignGeoTAXInfo can return up to 10 user-defined areas for each input address.

Table 34: Output Fields for User-Defined Boundary Files

Response Element	Max. Field Length (bytes)	Description
NumberUserBoundariesFound	3	The number of user-defined polygons found for the address.
UserBoundary*n*.BoundaryDescription	51 per User Boundary	A description of the polygon.
UserBoundary*n*.BoundaryID	11 per User Boundary	The ID of the polygon as specified in the user-defined boundary file.
UserBoundary*n*.BufferRelation	2 per User Boundary	Indicates where in the polygon the address resides in relation to the edge of the area. One of the following: **P** The address is inside the polygon at a distance from the edge that is greater than the specified

Response Element	Max. Field Length (bytes)	Description
		buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The address is inside the polygon but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The address is outside the polygon but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
UserBoundary*n*.DistanceToBorder	10 per User Boundary	Indicates the distance from the address to the border of the polygon. The distance is in the units specified by the option.
UserBoundary*n*.SupplementalBoundaryID	11 per User Boundary	A supplemental ID as specified in the user-defined boundary file.
UserBoundary*n*.BoundaryConfidence	4 per User Boundary	Provides the percentage overlap of the geocode to the User-defined boundary polygon layer. The returned percentage value describes the probability that the point falls in the User-defined boundary area.

Insurance Premium Tax District

The following table lists the output fields that contain Insurance Premium Tax District (IPD) data. For more information on insurance premium tax districts, see **AssignGeoTAXInfo** on page 70. To include IPD data in the output, set **Option.GeoTAXOutputRecordType** = I.

Note: AssignGeoTAXInfo returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 35: Insurance Premium Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberIPDsFound	3	The number of Insurance Premium Tax Districts found for the address
IPD*n*.BoundaryBuffer.BufferRelation	2 per IPD	Indicates where in the district the address resides in relation to the edge of the district. One of the following: **P** The address is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**.

Response Element	Max. Field Length (bytes)	Description
		I The address is inside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The address is outside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
IPD*n*.BoundaryBuffer.DistanceToBorder	10 per IPD	Indicates the distance from the address to the border of the district.
IPD*n*.BoundaryConfidence	4 per IPD	Provides the percentage overlap of the geocode to the IPD boundary polygon layer. The returned percentage value describes the probability that the point falls in the IPD boundary area.
IPD*n*.DistrictID	11 per IPD	IPD ID.
IPD*n*.DistrictName	61 per IPD	IPD name.
IPD*n*.DistrictType	7 per IPD	IPD district type.
IPD*n*.UpdateDate	7 per IPD	IPD update date (MMYYYY).
IPD*n*.VersionDate	7 per IPD	IPD compiled date (MMYYYY).
IPD*n*.Notes	21 per IPD	Tax code descriptions. For example: 01, 33, A, B
IPD*n*.ChangeDate	7 per IPD	IPD change date.
IPD*n*.EffectiveDate	7 per IPD	MMDDYY - Identifies when district becomes active - State supplied For example: 010108
IPD*n*.ExpirationDate	7 per IPD	MMDDYY - Identifies when district becomes inactive - State supplied For example: 063009
IPD*n*.FireRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.FireFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount

Response Element	Max. Field Length (bytes)	Description
		M - Multiple Percentages has a semi colon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.CasualtyRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.CasualtyFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.VehicleRate	21 per IPD	Format is dependent on associated flag Possible Values: .13, 15.00 or 3;7
IPD*n*.VehicleFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.MarineRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.MarineFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.HealthRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.HealthFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.LifeRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.LifeFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.OtherRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.OtherFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount

Response Element	Max. Field Length (bytes)	Description
		M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.MinimumRate	21 per IPD	Format is dependent on associated flag Possible Values: .13, 15.00 or 3;7
IPD*n*.MinimumFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"

Payroll Tax District

The following table lists the output fields that contain Payroll Tax District (PAY) data. For more information on payroll tax districts, see **AssignGeoTAXInfo** on page 70. To include this data in the output, set **Option.GeoTAXOutputRecordType** = R.

Note: AssignGeoTAXInfo returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 36: Payroll Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberPAYsFound	3	Number of PAYs returned.
PAY*n*.BoundaryBuffer.BufferRelation	2 per PAY	Indicates where in the district the address resides in relation to the edge of the district. One of the following: **P** The address is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The address is inside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The address is outside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
PAY*n*.BoundaryBuffer.DistanceToBorder	10 per PAY	Indicates the distance from the address to the border of the district. The distance is in the units specified by the option.

Response Element	Max. Field Length (bytes)	Description
PAY*n*.BoundaryConfidence	4 per PAY	Provides the percentage overlap of the geocode to the PAY boundary polygon layer. The returned percentage value describes the probability that the point falls in the PAY boundary area.
PAY*n*.DistrictID	11 per PAY	PAY district ID.
PAY*n*.DistrictName	61 per PAY	PAY district name.
PAY*n*.DistrictType	7 per PAY	PAY district type.
PAY*n*.ID	11 per PAY	PAY ID.
PAY*n*.MunicipalEMSTax	2 per PAY	PAY municipality emergency municipal services tax. The values for Pennsylvania are: **Y** Levies the tax **N** Does not levy the tax All other states are null.
PAY*n*.MunicipalIncomeTax	2 per PAY	PAY municipality income tax. The values for Pennsylvania are: **R** Resident **N** Non-resident **B** Both **X** None All other states are null.
PAY*n*.SchoolDistrictEMSTax	2 per PAY	PAY school district emergency municipal services tax. The Values for Pennsylvania are: **Y** Levies the tax **N** Does not levy the tax All other states are null.
PAY*n*.SchoolDistrictIncomeTax	2 per PAY	PAY school district income tax. The values for Pennsylvania are: **R** Resident **N** Non-resident **B** Both **X** N The values for Ohio are:

Response Element	Max. Field Length (bytes)	Description
		R Resident **X** None All other states are null.

Property Tax District

The following table lists the output fields that contain Property Tax District (PTD) data. For more information on property tax districts, see **AssignGeoTAXInfo** on page 70. To include this data in the output, set **Option.GeoTAXOutputRecordType** = P.

Note: AssignGeoTAXInfo returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 37: Property Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberPTDsFound	3	Number of PTDs returned.
PTD*n*.BoundaryBuffer.BufferRelation	2 per PTD	Indicates where in the district the address resides in relation to the edge of the district. One of the following: **P** The address is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The address is inside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The address is outside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
PTD*n*.BoundaryBuffer.DistanceToBorder	10 per PTD	Indicates the distance from the address to the border of the district. The distance is in the units specified by the option.
PTD*n*.BoundaryConfidence	4 per PTD	Provides the percentage overlap of the geocode to the PTD boundary polygon layer. The returned percentage value describes the probability that the point falls in the PTD boundary area.
PTD*n*.DistrictID	11 per PTD	PTD district ID.

Response Element	Max. Field Length (bytes)	Description
PTD*n*.DistrictName	61 per PTD	PTD district name.
PTD*n*.DistrictType	2 per PTD	PTD district type. Only returned for Indiana. **R** Reporting district **B** Billing district
PTD*n*.JurisdictionID	11 per PTD	PTD jurisdiction ID.
PTD*n*.UpdateDate	7 per PTD	PTD update date.

Special Purpose Tax District

The following table lists the output fields that contain Special Purpose Tax District (SPD) data. For more information on special purpose tax districts, see **AssignGeoTAXInfo** on page 70. To include this data in the output, set **Option.GeoTAXOutputRecordType** = S.

Note: AssignGeoTAXInfo returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 38: Special Purpose Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberSPDsFound	3	Number of SPDs returned.
SPD*n*.BoundaryBuffer.BufferRelation	2 per SPD	Indicates where in the district the address resides in relation to the edge of the district. One of the following: **P** The address is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The address is inside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The address is outside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
SPD*n*.BoundaryBuffer.DistanceToBorder	10 per SPD	Indicates the distance from the address to the border of the district. The distance is in the units specified by the option.

Response Element	Max. Field Length (bytes)	Description
SPD*n*.BoundaryConfidence	4 per SPD	Provides the percentage overlap of the geocode to the SPD boundary polygon layer. The returned percentage value describes the probability that the point falls in the SPD boundary area.
SPD*n*.CompiledDate	7 per SPD	SPD compiled date.
SPD*n*.DistrictCode	4 per SPD	3-digit district type code.
SPD*n*.DistrictName	61 per SPD	SPD name.
SPD*n*.DistrictNumber	6 per SPD	SPD district number.
SPD*n*.EffectiveDate	7 per SPD	SPD effective date.
SPD*n*.UpdateDate	7 per SPD	SPD update date.
SPD*n*.VersionDate	7 per SPD	SPD version date.

Sales and Use Tax Rates

The following table lists the output fields that contain the sales and use tax rate data.

To include tax rate data in the output, set **Option.GeoTAXOutputRecordType** = B.

To select the tax rate type, set **Option.TaxRate** to one of the following:

N	Do not return sales and use tax rates. (default)
G	Return the General sales and use tax rates.
A	Return the Automotive sales and use tax rates.
C	Return the Construction sales and use tax rates.
M	Return the Medical sales and use tax rates.

Note: You must be a licensed user of the Pitney Bowes Software Sales and Use Tax Rate file to use this feature.

Table 39: Sales and Use Tax Rate Output Fields

Response Element	Max. Field Length (bytes)	Description
TaxRate.RC	2	Tax Rate return code denoting the level of match obtained agained the Pitney Bowes Software Sales and Use Tax Rate file: **E** Exact match, using all 5 fields

Response Element	Max. Field Length (bytes)	Description
		P Partial match, using 4 fields **A** Alternate match, using 3 fields **N** Record is default-coded based on valid state code. **Blank** No matching PB Software Sales and Use Tax Rate record found.
Municipal.SalesTaxRate	11	Municipality sales tax rate for the selected tax rate type.
County.SalesTaxRate	11	County sales tax rate for the selected tax rate type.
State.SalesTaxRate	11	State sales tax rate for the selected tax rate type.
SPD*n*.SalesTaxRate	11 per SPD	Sales tax rate for up to 10 Special Purpose Districts (SPD).
TaxRate.SalesTotal	11	The sum of the individual Municipal, County, State and SPD sales tax rates.
Municipal.UseTaxRate	11	Municipality use tax rate for the selected tax rate type.
County.UseTaxRate	11	County use tax rate for the selected tax rate type.
State.UseTaxRate	11	State use tax rate for the selected tax rate type.
SPD*n*.UseTaxRate	11 per SPD	Use tax rate for up to 10 Special Purpose Districts (SPD).
TaxRate.UseTotal	11	The sum of the individual Municipal, County, State and SPD use tax rates.

Error Reporting

The following table defines the error reporting output fields.

Table 40: Error Output Fields

Response Element	Max. Field Length (bytes)	Description
GTX.ErrorCode	3	This field contains a return code if the GeoTAX engine experiences an abnormal termination. **Note:** This field contains the same set of codes returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform. The first character indicates the file (or set of files affected).

Response Element	Max. Field Length (bytes)	Description
		Blank Matcher terminated normally **A** User Auxiliary file problem **CE** coubsub.txb file problem **CI** Confidence engine problem **D** Boundary file **F** User-defined boundary file problem **G** Address Matching engine problem **L** Licensing problem **S** State file problem **U** GeoTAX Auxiliary file problem **X** Combination of Street and state file problem **Z** zip.gsb file problem The second position is one of the following: **E** Fatal issue, program terminating **F** Expired database **I** Informational
GTX.ErrorDescription	81	If the GeoTAX engine experiences an abnormal termination, this field contains a text description of the reason. It is blank if GeoTAX terminated normally. **Note:** This field contains the same set of descriptions returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform. SI-"TS158 FILES NOT FOUND" SI-"TS158 FILES VINTAGE OR INCOMPLETE DB ERROR" SI-"STATE FILES NOT FOUND" SE-"STATE AND TS158 FILES NOT FOUND" SE-"STATE NOT FOUND AND TS158 VINTAGE ERROR" SI-"STATE FILES VINTAGE OR INCOMPLETE DB ERROR" SE-"STATE VINTAGE ERROR AND TS158 NOT FOUND" SE-"STATE AND TS158 FILES VINTAGE OR INCOMPLETE DB ERROR" GI-"STREET FILES NOT FOUND" XI-"STREET AND TS158 FILES NOT FOUND" XI-"STREET NOT FOUND AND TS158 FILES VINTAGE ERROR"

Response Element	Max. Field Length (bytes)	Description
		XI-"STREET AND STATE FILES NOT FOUND" XE-"STREET STATE AND TS158 FILES NOT FOUND" XE-"STREET AND STATE NOT FOUND AND TS158 VINTAGE ERROR" XI-"STREET NOT FOUND AND STATE VINTAGE ERROR" XE-"STREET AND TS158 NOT FOUND AND STATE VINTAGE ERROR" XE-"STREET NOT FOUND AND STATE AND TS158 VINTAGE ERROR" GI-"STREET FILES VINTAGE OR INCOMPLETE DB ERROR" XI-"STREET VINTAGE ERROR AND TS158 NOT FOUND" XI-"STREET AND TS158 FILES VINTAGE OR INCOMPLETE DB ERROR" XI-"STREET VINTAGE ERROR AND STATE NOT FOUND" XE-"STREET VINTAGE ERROR AND STATE AND TS158 NOT FOUND" XE-"STREET AND TS158 VINTAGE ERROR AND STATE NOT FOUND" XI-"STREET AND STATE FILES VINTAGE OR INCOMPLETE DB ERROR" XE-"STREET AND STATE VINTAGE ERROR AND TS158 NOT FOUND" XE-"STREET STATE AND TS158 VINTAGE ERROR" LF-"INVALID FUNCTION PASSED TO GTDBLIO : " AI-"GENIO ERROR: FILE = G1GTAUX , FUNC = , ST = " UI-"GENIO ERROR: FILE = G1GTAX2 , FUNC = , ST = " XF-"The (DB Vintage) database has expired!" XF-"The (SPD file Vintage) SPD File has expired!" DI- "UNABLE TO VALIDATE BOUNDARY LICENSE" DI- "UNABLE TO OPEN BOUNDARY FILE" DI- "BOUNDARY FILE NOT FOUND" FI- "UNABLE TO VALIDATE USER BOUNDARY LICENSE" FI- "UNABLE TO OPEN USER BND FILE" FI- "USER BND FILE NOT FOUND"
GTX.WarnCode	3	This field contains warning codes returned by the GeoTAX engine. It is blank if no warnings were issued. A value of `WN` indicates a database will expire next month. **Note:** This field contains the same set of codes returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform.

Response Element	Max. Field Length (bytes)	Description
GTX.WarnDescription	81	A text description of any warnings returned by the GeoTAX engine. **Note:** This field contains the same set of descriptions returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform.
Status	2	Reports the success or failure of the match attempt. **null** Success **F** Failure. Some examples of failures are your license expired or you did not select any output record types and fields for AssignGeoTAXInfo to return.
Status.Code	12	If AssignGeoTAXInfo could not process the address, this field will show the reason. Currently there is one possible value for this field: Invalid Address.
Status.Description	64	If AssignGeoTAXInfo could not process the address, this field will show a description of the failure. One of the following: TS158 FILES NOT FOUND TS158 FILES VINTAGE OR INCOMPLETE DB ERROR STATE FILES NOT FOUND STATE AND TS158 FILES NOT FOUND STATE NOT FOUND AND TS158 VINTAGE ERROR STATE FILES VINTAGE OR INCOMPLETE DB ERROR STATE VINTAGE ERROR AND TS158 NOT FOUND STATE AND TS158 FILES VINTAGE OR INCOMPLETE DB ERROR STREET FILES NOT FOUND STREET AND TS158 FILES NOT FOUND STREET NOT FOUND AND TS158 FILES VINTAGE ERROR STREET AND STATE FILES NOT FOUND STREET STATE AND TS158 FILES NOT FOUND STREET AND STATE NOT FOUND AND TS158 VINTAGE ERROR STREET NOT FOUND AND STATE VINTAGE ERROR STREET AND TS158 NOT FOUND AND STATE VINTAGE ERROR STREET NOT FOUND AND STATE AND TS158 VINTAGE ERROR STREET FILES VINTAGE OR INCOMPLETE DB ERROR STREET VINTAGE ERROR AND TS158 NOT FOUND STREET AND TS158 FILES VINTAGE OR INCOMPLETE DB ERROR STREET VINTAGE ERROR AND STATE NOT FOUND

Response Element	Max. Field Length (bytes)	Description
		STREET VINTAGE ERROR AND STATE AND TS158 NOT FOUND STREET AND TS158 VINTAGE ERROR AND STATE NOT FOUND STREET AND STATE FILES VINTAGE OR INCOMPLETE DB ERROR STREET AND STATE VINTAGE ERROR AND TS158 NOT FOUND STREET STATE AND TS158 VINTAGE ERROR INVALID FUNCTION PASSED TO GTDBLIO : GENIO ERROR: FILE = G1GTAUX , FUNC = , ST = GENIO ERROR: FILE = G1GTAX2 , FUNC = , ST = The (DB Vintage) database has expired! The (SPD file Vintage) SPD File has expired! UNABLE TO VALIDATE BOUNDARY LICENSE UNABLE TO OPEN BOUNDARY FILE BOUNDARY FILE NOT FOUND UNABLE TO VALIDATE USER BOUNDARY LICENSE UNABLE TO OPEN USER BND FILE USER BND FILE NOT FOUND

CalculateDistance

CalculateDistance takes two sets of latitude/longitude coordinates as input, calculates the distance between the coordinates, and returns the distance between the two points.

CalculateDistance is part of the Enterprise Tax Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/CalculateDistance/results.json
```

XML endpoint:

```
http://server:port/rest/CalculateDistance/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/CalculateDistance/results.json?
Data.SecondLatitude=41.881833&Option.LatLongFormat=Decimal&
Data.SecondLongitude=-87.785587&Data.FirstLatitude=41.857333&
Data.FirstLongitude=-88.325183
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Distance": "27.799",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/CalculateDistance/results.xml?
Data.SecondLatitude=41.881833&Option.LatLongFormat=Decimal&
Data.SecondLongitude=-87.785587&Data.FirstLatitude=41.857333&
Data.FirstLongitude=-88.325183
```

The XML returned by this request would be:

```
<ns2:xml.CalculateDistanceResponse
xmlns:ns2="http://www.pb.com/spectrum/services/CalculateDistance">
   <ns2:output_port>
      <ns2:Result>
         <ns2:Distance>27.799</ns2:Distance>
         <ns2:user_fields/>
      </ns2:Result>
   </ns2:output_port>
</ns2:xml.CalculateDistanceResponse>
```

Request

Parameters for Input Data

CalculateDistance takes latitude and longitude information as input.

Table 41: CalculateDistance Input Data

Parameter	Description
Data.FirstLatitude	Latitude of the first point for which you want distance returned.
Data.FirstLatitude.Directional	First latitude directional. **N** North **S** South
Data.FirstLongitude	Longitude of the first point for which you want distance returned.
Data.FirstLongitude.Directional	First longitude directional. **E** East **W** West
Data.SecondLatitude	Latitude of the second point for which you want distance returned.
Data.SecondLatitude.Directional	Second latitude directional. **N** North **S** South
Data.SecondLongitude	Longitude of the second point for which you want distance returned.
Data.SecondLongitude.Directional	Second longitude directional. **E** East **W** West

Parameters for Options

Table 42: Output Data and Format Options

Parameter	Description
Option.LatLongFormat	Indicates the format of the input latitude/longitude. The options are: **DegMinSec** For example 90 00 00N180 00 00W. **PreZero** (090000000N180000000W). Default. **PreZeroDecimal** (090.000000N180.000000W) **Decimal** (90.000000-180.000000) **DecimalAssumed** (90000000-180000000)
Option.ReturnUnits	Indicates the measurement units returned for distance calculation: • Miles • Km • Feet • Meters

Response

CalculateDistance always returns the Confidence field to indicate the confidence in the output provided.

If CalculateDistance fails to process the data, it returns the fields Status, Status.Code, and Status.Descriptions. These fields provide information on why CalculateDistance failed to process the data. Some examples of failures are your license expired or you did not select any output record types and fields for CalculateDistance to return. The following table provides the record-level qualifiers and data outputs for CalculateDistance.

Table 43: CalculateDistance Output Fields

Response Element	Max. Field Length (bytes)	Description
Distance	9	Distance between the two input coordinates in the units of measurement that you specified.
Status	2	Reports the success or failure of the match attempt: **null** Success **F** Failure
Status.Code	2	Reason for failure or error. If Status = F, Status.Code = Failure.
Status.Description	64	Description of the problem. If Status = F, Status.Description = Unable to compute distance.

ReverseGeoTAXInfoLookup

ReverseGeoTAXInfoLookup allows latitude/longitude coordinates to be supplied as input and identifies the tax districts that apply to the given coordinate. Specifically, ReverseGeoTAXInfoLookup can return the following information about a location:

- FIPS state codes and county codes
- State and county names
- MCD codes and names
- Place codes and names
- Boundary file districts
- Cross-reference tax keys
- The relationship of the input coordinates to user-defined polygons
- Sales and use tax rates, if licensed for the Pitney Bowes Software Sales and Use Tax Rate File

ReverseGeoTAXInfoLookup optionally includes enhanced tax jurisdiction information for a location, including:

- **Insurance premium districts**—Areas designated for the collection of taxes imposed on insurance policy premiums, based on the policy holder's address. Insurance premium districts are created by state governments.
- **Payroll tax districts**—Areas designated for the collection of taxes imposed on employers to support state or local government facilities and services, based on the employee's and/or employer's address. Examples include taxes collected for districts to pay for schools, police, or other services. Payroll tax districts are created by state or local governments.
- **Payroll system tax codes**—Codes that represent specific jurisdictions that collect payroll tax. Using payroll system tax codes has advantages over using the payroll tax district information returned by ReverseGeoTAXInfoLookup:

  - ReverseGeoTAXInfoLookup uses an additional database to determine payroll tax codes, resulting in more accurate payroll tax determination.
  - Many payroll systems use specific codes to determine withholding amounts. Since you can customize the payroll tax codes returned by ReverseGeoTAXInfoLookup, you can set up a process where ReverseGeoTAXInfo Lookup returns the exact payroll tax codes required by your payroll system, instead of returning jurisdictional IDs that must then be translated into the codes used by your system.

- **Property tax districts**—Areas designated for the collection of taxes imposed on property owners to support local government facilities and services, based on the property's location. Examples include taxes collected for districts to pay for schools, police, or other services. Property tax districts are created by local governments.
- **Special purpose tax districts**—Areas designated for the collection of taxes imposed on residents to support specialized services for residents of the district, based on the resident's address. Examples include services such as sewer service, transit service, or water resources. Special purpose tax districts are created by legislative action, court action, or public referendums. This optional information requires the use of boundary files which require an additional license. Contact your Pitney Bowes Software sales representative for more information.

Using the optional Pitney Bowes Software Sales and Use Tax Rate file, ReverseGeoTAXInfoLookup includes tax rate data for a location, including:

Tax rate type:

- General
- Automotive
- Medical
- Construction

Sales and/or use tax rates for:

- State
- County
- Municipality
- Up to 10 SPDs

- Total Rate - the sum of the individual state, county, municipality and SPD rates.

Required input format

The required format for the input coordinates is as follows:

Table 44: Required Input Latitude/Longitude Format

Response Element	Format
Data.InputLatitude	00.000000 or without the decimal point 00000000
Data.InputLongitude	000.000000 or without the decimal point 000000000, or 00.000000 or without the decimal point 00000000

ReverseGeoTAXInfoLookup is part of the Enterprise Tax Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/ReverseGeoTAXInfoLookup/results.json
```

XML endpoint:

```
http://server:port/rest/ReverseGeoTAXInfoLookup/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/ReverseGeoTAXInfoLookup/results.json
?Data.InputLatitude=40.018998&Data.InputLongitude=-105.239580
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Confidence": "100.0",
   "ProcessedBy": "GTX",
   "County.Code": "013",
   "County.Name": "Boulder",
   "StateCode": "08",
   "InputLatitude": "40.018998",
   "InputLongitude": "-105.239580",
   "State.Abbreviation": "CO",
   "Place.ClassCode": "C1",
   "Place.Code": "07850",
   "Place.IncorporatedFlag": "Inc",
   "Place.Name": "Boulder",
   "Place.LastAnnexedDate": "10/2011",
   "Place.LastUpdatedDate": "04/2013",
   "Place.LastVerifiedDate": "01/2013",
   "Place.DistanceToBorder": "000000387",
   "Place.PointStatus": "P",
   "GNISCode": "002409883",
   "GTX.ErrorCode": "",
   "GTX.ErrorDescription": "",
   "GTX.WarnCode": "",
   "GTX.WarnDescription": ""
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/ReverseGeoTAXInfoLookup/results.xml
?Data.InputLatitude=40.018998&Data.InputLongitude=-105.239580
```

The XML returned by this request would be:

```
ns2:xml.ReverseGeoTAXInfoLookup
xmlns:ns2="http://www.pb.com/spectrum/services/ReverseGeoTAXInfoLookup">
   <ns2:output_port>
      <ns2:Address>
         <ns2:Confidence>100.0</ns2:Confidence>
         <ns2:ProcessedBy>GTX</ns2:ProcessedBy>
         <ns2:County.Code>013</ns2:County.Code>
         <ns2:County.Name>Boulder</ns2:County.Name>
         <ns2:StateCode>08</ns2:StateCode>
         <ns2:InputLatitude>40.018998</ns2:InputLatitude>
         <ns2:InputLongitude>-105.239580</ns2:InputLongitude>
         <ns2:State.Abbreviation>CO</ns2:State.Abbreviation>
         <ns2:Place.ClassCode>C1</ns2:Place.ClassCode>
         <ns2:Place.Code>07850</ns2:Place.Code>
         <ns2:Place.IncorporatedFlag>Inc</ns2:Place.IncorporatedFlag>
         <ns2:Place.Name>Boulder</ns2:Place.Name>
         <ns2:Place.LastAnnexedDate>10/2011</ns2:Place.LastAnnexedDate>
         <ns2:Place.LastUpdatedDate>04/2013</ns2:Place.LastUpdatedDate>
         <ns2:Place.LastVerifiedDate>01/2013</ns2:Place.LastVerifiedDate>
        <ns2:Place.DistanceToBorder>000000387</ns2:Place.DistanceToBorder>

         <ns2:Place.PointStatus>P</ns2:Place.PointStatus>
         <ns2:GNISCode>002409883</ns2:GNISCode>
         <ns2:GTX.ErrorCode>""</ns2:GTX.ErrorCode>
         <ns2:GTX.ErrorDescription>""</ns2:GTX.ErrorDescription>
         <ns2:GTX.WarnCode>""</ns2:GTX.WarnCode>
         <ns2:GTX.WarnDescription>""</ns2:GTX.WarnDescription>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.ReverseGeoTAXInfoLookup>
```

Request

Geocoding Options

Reverse geocoding information lookup is the process of taking an input latitude/longitude coordinate and returning jurisdictional tax information. The geocoding options define the database to use for matching and specify the distance units and buffer distance to use when matching to a boundary file.

Table 45: ReverseGeoTAXInfoLookup Geocoding Options

Parameter	Description
Option.Database.GTX	Select the database resource to use in the reverse geocoding lookup process.
Boundary matching: The following options can be set when matching to a boundary file such as SPD, IPD, PAY, Place and MCD, or user-defined.	
Option.DistanceUnits	Specifies the units in which to measure distance. One of the following: **Feet** Distances are measured in feet. (Default) **Km** Distances are measured in kilometers. **Meters** Distances are measured in meters.

Parameter	Description
	Miles Distances are measured in miles.
Option.DefaultBufferWidth	Specifies the buffer width to use for tax district boundary files. The tax district boundary files are the Special Purpose District (SPD) file, the Insurance Premium District (IPD) file, the Payroll Tax District (PAY) file, the Personal Property Tax District (PTD) file, Place and MCD files. The default buffer width that you specify here can be overridden on a record-by-record basis using the BufferWidth input field. For more information on buffers, see **Buffering** on page 290.
Option.DefaultUserBufferWidth	Specifies the buffer width to use for user-defined boundary files. Specify the distance in the units of measurement specified in the **Distance units** option. For information on buffers, see **Buffering** on page 290. The default buffer width that you specify here can be overridden on a record-by-record basis using the BufferWidth input field. **Note:** To use buffers, the user-defined boundary file must support buffers.

Output Data Options

Data options control the data returned by ReverseGeoTAXInfoLookup. **Table 46: ReverseGeoTAXInfo Lookup Data Options** on page 114 lists the output data options.

Table 46: ReverseGeoTAXInfo Lookup Data Options

Parameter	Description
Option.GeoTAXOutputRecordType	Select one or more of the following to obtain the type of data you want returned. ReverseGeoTAXInfo Lookup groups the output fields into record types. If you do not want all of the fields in a record type returned, do not select the check box, and list only those fields you want returned in Extra Output Fields. • **C**—Census • **T**—Tax Jurisdiction • **U**—User-defined boundary file • **W**—Payroll System Tax Codes • **X**—Auxiliary File • **B**—PB Software Sales and Use Tax Rate file You can also specify one, and only one, of the following: **I** Insurance Premium Tax District (IPD) **P** Property Tax District (PTD) **R** Payroll Tax District (PAY) **S** Special Purpose Tax District (SPD) For a description of the fields in each output group, see **Response** on page 116.

Parameter	Description
	Note: If you specify `W`, to obtain the best payroll system tax code match possible.
Option.TaxKey	If you integrate ReverseGeoTAXInfo Lookup with third-party tax compliance software from Vertex or Taxware, select which vendor you use. This controls the value returned in the GeoTAXKey output field. One of the following: **T** Return the Taxware jurisdiction code for the address. **Y** Return the Vertex jurisdiction code for the address. Select this option if you obtained a MatchMaster file from Vertex. **V** Return the Vertex jurisdiction code for the address. Select this option if you obtained a Vertex file from Pitney Bowes Software. **N** Do not return either the Taxware or Vertex jurisdiction codes (default).
Option.TaxRate	Select the desired Sales tax rate type or None: **N** Do not return sales tax rates. (default) **G** Return the General sales tax rates. **A** Return the Automotive sales tax rates. **C** Return the Construction sales tax rates. Return the Medical sales tax rates.
Option.OutputFields	Indicates the individual output fields you want returned. You can use this field instead of the Output Record Type to limit the output to those fields that are important to your current data needs. For a list of the fields included in each data type, see **Response** on page 116.

Output Format

Output format options control how ReverseGeoTAXInfo Lookup formats output data. The following table lists the output format options.

Table 47: ReverseGeoTAXInfoLookup Output Format Options

Parameter	Description
Option.OutputCasing	Specifies the casing of these output fields: County.Name, MCD.Name, Place.Name, IPDn.DistrictName, PAYn.DistrictName, SPDn.DistrictName, and PTCn.PayrollDescription. One of the following: **M** The output in mixed case (default). For example: Boulder. **U** The output in upper case. For example: BOULDER.

Response

Auxiliary File

The following table lists the output fields that contain auxiliary file data. To include auxiliary file data in the output, set **Option.GeoTAXOutputRecordType** = X. The following table lists the output fields that contain tax jurisdiction data.

Table 48: Auxiliary File Output Fields

Response Element	Max. Field Length (bytes)	Description
AuxiliaryData.AuxiliaryFile	301	Data retrieved as a result of an auxiliary match from the user-defined area of the auxiliary file.
AuxiliaryData.StateFile	201	Data retrieved as a result of a state match. Data content and format vary depending on the state file used.

Census

The census output fields contains census information from the U.S. Census, including Minor Civil Divisions (MCDs) and Census County Division (CCD) names and codes. MCDs are the primary political or administrative divisions of a county, representing many kinds of legal entities with a variety of governmental and administrative functions. CCDs are established in states where there are no legally established MCDs. The Census Bureau recognizes MCDs in 28 states and has established CCDs in 21 states. The District of Columbia has no primary divisions, and the city of Washington, DC is considered equivalent to an MCD for data presentation purposes.

Census data also contains the Federal Information Processing Standards (FIPS) codes for each state and county. The FIPS State Code and the FIPS County Code are both used by the Census Bureau to identify these geographic units.

The following table lists the output fields that contain census data. To include census data in the output, set **Option.GeoTAXOutputRecordType** = C.

Table 49: Census Data Output Fields

Response Element	Max. Field Length (bytes)	Description
County.Code	4	Three-digit Federal Information Processing Standards (FIPS) county code extracted from the Census.BlockCode. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output.
County.Name	26	Name of the county. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output.

Response Element	Max. Field Length (bytes)	Description
MCD.Code	6	Minor Civil Division/Census County Division (MCD/CCD) Code.
MCD.Name	41	Minor Civil Division/Census County Division (MCD/CCD) name.
MCD.PointStatus	2	An address can be compared to a Minor Civil Division (MCD)/county subdivision file (Cousub.txt). This output field returns the result for a comparison between the matched geocode location to the polygon defined by the Cousub.txb file. For more information on buffers, see **Buffering** on page 290. **Note:** The buffer distance for Cousub.txb is internally set to zero and cannot be modified. **P** The point is in the polygon. **I** The point is in the buffer area inside the polygon. **B** The point is in the buffer area and outside of the polygon. **blank** Polygon not found.
MCD.DistanceToBorder	10	Returns the distance in feet between the matched address point to the polygon defined by the Cousub.txb file.
StateCode	3	Two-digit Federal Information Processing Standards (FIPS) state code extracted from the Census.BlockCode. **Note:** This field is always included in the output regardless of whether or not you choose to include census data in the output.

Input Latitude/Longitude

ReverseGeoTAXInfoLookup always returns the input coordinates as part of the output. The input latitude/longitude fields are returned as input from the data. ReverseGeoTAXInfoLookup does not change these input values.

Table 50: Input Latitude/Longitude Output Fields

Response Element	Max. Field Length (bytes)	Description
InputLatitude	12	Input latitude.
InputLongitude	12	Input longitude.

Payroll System Tax Code

The following table lists the output fields that contain Payroll System Tax Code (PTC) data. For more information on payroll tax districts, see **Reverse GeoTAX Info Lookup**. To include this data in the output, set **Option.GeoTAXOutputRecordType** = W.

Note: ReverseGeoTAXInfoLookup returns up to six payroll tax codes per input location.

Table 51: Payroll System Tax Code Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberPTCsFound	2	The number of payroll system tax codes found for this location.
PTC*n*.MatchCode	2 per PTC	Indicates the level of match obtained for the location. In order from most specific match to least, the possible match codes are: **P** The address was matched to a specific Payroll District ID. This is the most specific match. **G** The address was matched to a GNIS Code. **F** The address was matched to a county's FIPS code. **S** The address was matched to a state's FIPS code. This is the least specific match.
PTC*n*.PayrollCode	16 per PTC	A code that represents a taxing authority in a payroll application. This is a user-defined code. The specific codes are determined by the payroll application that utilizes the data returned by ReverseGeoTAXInfo Lookup.
PTC*n*.PayrollDescription	41 per PTC	A description of the purpose of this payroll code.
PTC*n*.PayrollFlag	7 per PTC	A user-defined flag from the PTC database.
StateCounty	33	The state abbreviation and county name.

Tax Jurisdiction

Tax jurisdiction data contains information about the "place" for the location. A "place" is a geographic area defined on the basis of population criteria that vary by state; or, an area recognized as significant because it is located in an incorporated municipality. Places are used to determine tax jurisdiction.

The following table lists the output fields that contain tax jurisdiction data. To include tax jurisdiction data in the output, set **Option.GeoTAXOutputRecordType** = T.

Table 52: Tax Jurisdiction Output Fields

Response Element	Max. Field Length (bytes)	Description
GeoTAXKey	10	The value in this field varies depending on the option you specified in the Option.TaxKey option: If you specified `T`, GeoTAXKey contains the proprietary codes used in Taxware tax compliance software. You can use this code in your Taxware application to find out the tax rate for the jurisdiction. If you specified `Y` or `V`, GeoTAXKey contains the proprietary Vertex® jurisdiction code (comprised of a two-digit Vertex® state code, three-digit FIPS county code, and four-digit Vertex® city code). You can use this code in your Vertex® application to find out the tax rate for the jurisdiction.
GeoTAXKey.MatchCode	2	Return code denoting the level of match obtained against the Vertex or Taxware cross reference files. **E** Exact match using five fields: FIPS state code, FIPS county code, FIPS or GNIS place code, ZIP Code, and FIPS place name. **P** Partial match using four fields: FIPS state code, FIPS county code, FIPS or GNIS place code, and ZIP Code. **A** Alternate match using two fields: ZIP Code, FIPS place name. **N** Record is default coded based on valid state code. **null** No matching record found.
GeoTAXKey.MatchLevel	12	A description of the value returned in the GeoTAXKey.MatchCode field. **Exact** Exact match. See description in GeoTAXKey.MatchCode. **Partial** Partial match. See description in GeoTAXKey.MatchCode. **Alternate** Alternate match. See description in GeoTAXKey.MatchCode. **DefaultCode** Record is default coded. See description in GeoTAXKey.MatchCode. **NoMatch** No matching record found.
GNISCode	10	Unique nine-digit Geographic Names Information System (GNIS) code.

Response Element	Max. Field Length (bytes)	Description
		Note: This field is always included in the output regardless of whether or not you choose to include census data in the output.
Place.ClassCode	3	Place class code. Place class codes are used to determine the proper taxing jurisdictions **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.Code	6	An identifier for a specific place. A "place" is a geographic area defined on the basis of population criteria that vary by state. Or, an area recognized as significant because it is located in an incorporated municipality. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.IncorporatedFlag	8	Indicates whether the address is located in an incorporated or unincorporated place. A "place" is a geographic area defined on the basis of population criteria that vary by state. Or, an area recognized as significant because it is located in an incorporated municipality. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output. **Inc** Incorporated place code. **Uninc** Unincorporated place code. **Unknown** Incorporation status unknown.
Place.LastAnnexedDate	8	Last annexed date, in the format MM/YYYY, representing the month and year of the most recent boundary change or the most recent available boundary information. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.LastUpdatedDate	8	Last updated date, in the format MM/YYYY, reflecting the month and year when TomTom updated the database to reflect attribute (name change, FIPS change, etc.) or boundary edits to the Place. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.

Response Element	Max. Field Length (bytes)	Description
Place.LastVerifiedDate	8	Last verified date, in the format MM/YYYY, representing the month and year that TomTom verified municipality change information. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.Name	41	The name of the "place" for the location. A "place" is a geographic area defined on the basis of population criteria that vary by state. Or, an area recognized as significant because it is located in an incorporated municipality. **Note:** This field is always included in the output regardless of whether or not you choose to include tax jurisdiction data in the output.
Place.PointStatus	2	Returns the result for a comparison between the matched geocode location to the polygon defined by the Place.txb file. For more information on buffers, see **Buffering** on page 290. **P** The point is in the polygon. **I** The point is in the buffer area inside the polygon. **B** The point is in the buffer area and outside of the polygon. **blank** Polygon not found.
Place.DistanceToBorder	10	Returns the distance in feet between the matched address point to the polygon defined by the Place.txb file.

User-Defined Boundary File

The following table lists the output fields that contain data returned from user-defined boundary files. To include this data in the output, set **Option.GeoTAXOutputRecordType** = U.

Note: ReverseGeoTAXInfoLookup can return up to 10 user-defined areas for each input location.

Table 53: Output Fields for User-Defined Boundary Files

Response Element	Max. Field Length (bytes)	Description
NumberUserBoundariesFound	3	The number of user-defined polygons found for the address.

Response Element	Max. Field Length (bytes)	Description
UserBoundary*n*.BoundaryDescription	51 per User Boundary	A description of the polygon.
UserBoundary*n*.BoundaryID	11 per User Boundary	The ID of the polygon as specified in the user-defined boundary file.
UserBoundary*n*.BufferRelation	2 per User Boundary	Indicates where in the polygon the location resides in relation to the edge of the area. One of the following: **P** The geocode is inside the polygon at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The geocode is inside the polygon but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The geocode is outside the polygon but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
UserBoundary*n*.DistanceToBorder	10 per User Boundary	Indicates the distance in feet from the input location to the border of the polygon.
UserBoundary*n*.SupplementalBoundaryID	11 per User Boundary	A supplemental ID as specified in the user-defined boundary file.

Insurance Premium Tax Districts

The following table lists the output fields that contain Insurance Premium Tax Districts (IPD) data. For more information on insurance premium tax districts, see **ReverseGeoTAXInfoLookup** on page 111. To include IPD data in the output, set **Option.GeoTAXOutputRecordType** = I.

Note: ReverseGeoTAXInfoLookup returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 54: Insurance Premium Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberIPDsFound	3	The number of Insurance Premium Tax Districts found for the location.

Response Element	Max. Field Length (bytes)	Description
IPD*n*.BoundaryBuffer.BufferRelation	2 per IPD	Indicates where in the district the location resides in relation to the edge of the district. One of the following: **P** The location is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The location is inside the district but is close to the edge. This indicates that the location is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The location is outside the district but is close to the edge. This indicates that the location is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
IPD*n*.BoundaryBuffer.DistanceToBorder	10 per IPD	Indicates the distance in feet from the location to the border of the district.
IPD*n*.DistrictID	11 per IPD	IPD ID.
IPD*n*.DistrictName	61 per IPD	IPD name.
IPD*n*.DistrictType	7 per IPD	IPD district type.
IPD*n*.UpdateDate	7 per IPD	IPD update date (MMYYYY).
IPD*n*.VersionDate	7 per IPD	IPD compiled date (MMYYYY).
IPD*n*.Notes	21 per IPD	Tax code descriptions. For example: 01, 33, A, B
IPD*n*.ChangeDate	7 per IPD	IPD change date.
IPD*n*.EffectiveDate	7 per IPD	MMDDYY - Identifies when district becomes active - State supplied For example: 010108
IPD*n*.ExpirationDate	7 per IPD	MMDDYY - Identifies when district becomes inactive - State supplied For example: 063009
IPD*n*.FireRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7

Response Element	Max. Field Length (bytes)	Description
IPD*n*.FireFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semi colon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.CasualtyRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.CasualtyFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.VehicleRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.VehicleFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.MarineRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.MarineFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.HealthRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.HealthFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.LifeRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.LifeFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.OtherRate	21 per IPD	Format is dependent on associated flag Possible Values: .13, 15.00 or 3;7

Response Element	Max. Field Length (bytes)	Description
IPD*n*.OtherFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"
IPD*n*.MinimumRate	21 per IPD	Format is dependent on associated flag For example: .13, 15.00 or 3;7
IPD*n*.MinimumFlag	6 per IPD	P - Percentage; .1 = 10%, .0575 = 5.75% F - Flat Fee dollar amount M - Multiple Percentages has a semicolon as a delimiter. 3;7 = "3% or 7%"

Payroll Tax Districts

The following table lists the output fields that contain Payroll Tax District (PAY) data. For more information on payroll tax districts, see **Reverse GeoTAX Info Lookup**. To include this data in the output, set **Option.GeoTAXOutputRecordType** = R.

Note: ReverseGeoTAXInfoLookup returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 55: Payroll Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberPAYsFound	3	Number of payroll tax districts found for the location.
PAY*n*.BoundaryBuffer.BufferRelation	2 per PAY	Indicates where in the district the location resides in relation to the edge of the district. One of the following: **P** The location is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The location is inside the district but is close to the edge. This indicates that the location is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The location is outside the district but is close to the edge. This indicates that the location is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.

Response Element	Max. Field Length (bytes)	Description
PAY*n*.BoundaryBuffer.DistanceToBorder	10 per PAY	Indicates the distance in feet from the location to the border of the district.
PAY*n*.DistrictID	11 per PAY	PAY district ID.
PAY*n*.DistrictName	61 per PAY	PAY district name.
PAY*n*.DistrictType	7 per PAY	PAY district type.
PAY*n*.ID	11 per PAY	PAY ID.
PAY*n*.MunicipalEMSTax	2 per PAY	PAY municipality emergency municipal services tax. The values for Pennsylvania are: **Y** Levies the tax **N** Does not levy the tax All other states are null.
PAY*n*.MunicipalIncomeTax	2 per PAY	PAY municipality income tax. The values for Pennsylvania are: **R** Resident **N** Non-resident **B** Both **X** None All other states are null.
PAY*n*.SchoolDistrictEMSTax	2 per PAY	PAY school district emergency municipal services tax. The Values for Pennsylvania are: **Y** Levies the tax **N** Does not levy the tax All other states are null.
PAY*n*.SchoolDistrictIncomeTax	2 per PAY	PAY school district income tax. The values for Pennsylvania are: **R** Resident **N** Non-resident **B** Both **X** N The values for Ohio are: **R** Resident **X** None

Response Element	Max. Field Length (bytes)	Description
		All other states are null.

Property Tax District

The following table lists the output fields that contain Property Tax District (PTD) data. For more information on property tax districts, see **ReverseGeoTAXInfoLookup** on page 111. To include this data in the output, set **Option.GeoTAXOutputRecordType** = P.

Note: ReverseGeoTAXInfoLookup returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 56: Property Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberPTDsFound	3	The number of Property Tax Districts found for the location.
PTD*n*.BoundaryBuffer.BufferRelation	2 per PTD	Indicates where in the district the location resides in relation to the edge of the district. One of the following: **P** The location is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The location is inside the district but is close to the edge. This indicates that the location is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The location is outside the district but is close to the edge. This indicates that the address is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
PTD*n*.BoundaryBuffer.DistanceToBorder	10 per PTD	Indicates the distance in feet from the location to the border of the district.
PTD*n*.DistrictID	11 per PTD	PTD district ID.
PTD*n*.DistrictName	61 per PTD	PTD district name.
PTD*n*.DistrictType	2 per PTD	PTD district type. Only returned for Indiana. **R** Reporting district **B** Billing district

Response Element	Max. Field Length (bytes)	Description
PTD*n*.JurisdictionID	11 per PTD	PTD jurisdiction ID.
PTD*n*.UpdateDate	7 per PTD	PTD update date.

Special Purpose Tax Districts

The following table lists the output fields that contain Special Purpose Tax Districts (SPD) data. For more information on special purpose tax districts, see **Reverse GeoTAX Info Lookup**. To include this data in the output, set **Option.GeoTAXOutputRecordType** = S.

Note: ReverseGeoTAXInfoLookup returns multiple districts for IPDs, SPDs, PTDs, and PAYs.

Table 57: Special Purpose Tax District Output Fields

Response Element	Max. Field Length (bytes)	Description
NumberSPDsFound	3	Number of Special Purpose Tax Districts found for the location.
SPD*n*.BoundaryBuffer.BufferRelation	2 per SPD	Indicates where in the district the location resides in relation to the edge of the district. One of the following: **P** The location is inside the district at a distance from the edge that is greater than the specified buffer width. Buffer width is specified either by the option or by the input field **BufferWidth**. **I** The location is inside the district but is close to the edge. This indicates that the location is in the buffer area specified either by the option or by the input field **BufferWidth**. **B** The location is outside the district but is close to the edge. This indicates that the location is in the buffer area specified either by the option or by the input field **BufferWidth**. For more information, see **Buffering** on page 290.
SPD*n*.BoundaryBuffer.DistanceToBorder	10 per SPD	Indicates the distance in feet from the address to the border of the district.
SPD*n*.CompiledDate	7 per SPD	SPD compiled date.
SPD*n*.DistrictCode	4 per SPD	3-digit district type code.

Response Element	Max. Field Length (bytes)	Description
SPD*n*.DistrictName	61 per SPD	SPD name.
SPD*n*.DistrictNumber	6 per SPD	SPD district number.
SPD*n*.EffectiveDate	7 per SPD	SPD effective date.
SPD*n*.UpdateDate	7 per SPD	SPD update date.
SPD*n*.VersionDate	7 per SPD	SPD version date.

Sales and Use Tax Rates

The following table lists the output fields that contain the sales and use tax rate data.

To include tax rate data in the output, set **Option.GeoTAXOutputRecordType** = B.

To select the tax rate type, set **Option.TaxRate** to one of the following:

N	Do not return sales and use tax rates. (default)
G	Return the General sales and use tax rates.
A	Return the Automotive sales and use tax rates.
C	Return the Construction sales and use tax rates.
M	Return the Medical sales and use tax rates.

Note: You must be a licensed user of the Pitney Bowes Software Sales and Use Tax Rate file to use this feature.

Table 58: Sales and Use Tax Rate Output Fields

Response Element	Max. Field Length (bytes)	Description
TaxRate.RC	2	Tax Rate return code denoting the level of match obtained agained the Pitney Bowes Software Sales and Use Tax Rate file: **E** Exact match, using all 5 fields **P** Partial match, using 4 fields **A** Alternate match, using 3 fields **N** Record is default-coded based on valid state code. **Blank** No matching PB Software Sales and Use Tax Rate record found.

Response Element	Max. Field Length (bytes)	Description
Municipal.SalesTaxRate	11	Municipality sales tax rate for the selected tax rate type.
County.SalesTaxRate	11	County sales tax rate for the selected tax rate type.
State.SalesTaxRate	11	State sales tax rate for the selected tax rate type.
SPD*n*.SalesTaxRate	11 per SPD	Sales tax rate for up to 10 Special Purpose Districts (SPD).
TaxRate.SalesTotal	11	The sum of the individual Municipal, County, State and SPD sales tax rates.
Municipal.UseTaxRate	11	Municipality use tax rate for the selected tax rate type.
County.UseTaxRate	11	County use tax rate for the selected tax rate type.
State.UseTaxRate	11	State use tax rate for the selected tax rate type.
SPD*n*.UseTaxRate	11 per SPD	Use tax rate for up to 10 Special Purpose Districts (SPD).
TaxRate.UseTotal	11	The sum of the individual Municipal, County, State and SPD use tax rates.

Error Reporting

The following table defines the error reporting output fields.

Table 59: Error Output Fields

Response Element	Max. Field Length (bytes)	Description
GTX.ErrorCode	3	This field contains a return code if the GeoTAX engine experiences an abnormal termination. **Note:** This field contains the same set of codes returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform. The first character indicates the file (or set of files affected). **Blank** Matcher terminated normally **A** User Auxiliary file problem **CE** coubsub.txb file problem **CI** Confidence engine problem **D** Boundary file **F** User-defined boundary file problem

Response Element	Max. Field Length (bytes)	Description
		G Address Matching engine problem **L** Licensing problem **S** State file problem **U** GeoTAX Auxiliary file problem **X** Combination of Street and state file problem **Z** zip.gsb file problem The second position is one of the following: **E** Fatal issue, program terminating **F** Expired database **I** Informational
GTX.ErrorDescription	81	If the GeoTAX engine experiences an abnormal termination, this field contains a text description of the reason. It is blank if GeoTAX terminated normally. The maximum length is 80. **Note:** This field contains the same set of descriptions returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform. SI-"TS158 FILES NOT FOUND" SI-"TS158 FILES VINTAGE OR INCOMPLETE DB ERROR" SI-"STATE FILES NOT FOUND" SE-"STATE AND TS158 FILES NOT FOUND" SE-"STATE NOT FOUND AND TS158 VINTAGE ERROR" SI-"STATE FILES VINTAGE OR INCOMPLETE DB ERROR" SE-"STATE VINTAGE ERROR AND TS158 NOT FOUND" SE-"STATE AND TS158 FILES VINTAGE OR INCOMPLETE DB ERROR" GI-"STREET FILES NOT FOUND" XI-"STREET AND TS158 FILES NOT FOUND" XI-"STREET NOT FOUND AND TS158 FILES VINTAGE ERROR" XI-"STREET AND STATE FILES NOT FOUND" XE-"STREET STATE AND TS158 FILES NOT FOUND" XE-"STREET AND STATE NOT FOUND AND TS158 VINTAGE ERROR" XI-"STREET NOT FOUND AND STATE VINTAGE ERROR" XE-"STREET AND TS158 NOT FOUND AND STATE VINTAGE ERROR"

Response Element	Max. Field Length (bytes)	Description
		XE-"STREET NOT FOUND AND STATE AND TS158 VINTAGE ERROR" GI-"STREET FILES VINTAGE OR INCOMPLETE DB ERROR" XI-"STREET VINTAGE ERROR AND TS158 NOT FOUND" XI-"STREET AND TS158 FILES VINTAGE OR INCOMPLETE DB ERROR" XI-"STREET VINTAGE ERROR AND STATE NOT FOUND" XE-"STREET VINTAGE ERROR AND STATE AND TS158 NOT FOUND" XE-"STREET AND TS158 VINTAGE ERROR AND STATE NOT FOUND" XI-"STREET AND STATE FILES VINTAGE OR INCOMPLETE DB ERROR" XE-"STREET AND STATE VINTAGE ERROR AND TS158 NOT FOUND" XE-"STREET STATE AND TS158 VINTAGE ERROR" LF-"INVALID FUNCTION PASSED TO GTDBLIO : " AI-"GENIO ERROR: FILE = G1GTAUX , FUNC = , ST = " UI-"GENIO ERROR: FILE = G1GTAX2 , FUNC = , ST = " XF-"The (DB Vintage) database has expired!" XF-"The (SPD file Vintage) SPD File has expired!" DI- "UNABLE TO VALIDATE BOUNDARY LICENSE" DI- "UNABLE TO OPEN BOUNDARY FILE" DI- "BOUNDARY FILE NOT FOUND" FI- "UNABLE TO VALIDATE USER BOUNDARY LICENSE" FI- "UNABLE TO OPEN USER BND FILE" FI- "USER BND FILE NOT FOUND"
GTX.WarnCode	3	This field contains warning codes returned by the GeoTAX engine. It is blank if no warnings were issued. A value of `WN` indicates a database will expire next month. **Note:** This field contains the same set of codes returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform.
GTX.WarnDescription	81	A text description of any warnings returned by the GeoTAX engine. **Note:** This field contains the same set of descriptions returned by the standalone GeoTAX software and is intended for users who have migrated from GeoTAX to Spectrum™ Technology Platform.

Match and Location Codes

Match Codes

The following table lists the Match Codes. The Match Codes indicate the portions of the address that matched or did not match to the reference file. If a match could not be made, the Match Code begins with E and the remaining digits indicate why the address did not match. The digits do not specifically refer to which address elements did not match, but rather why the address did not match. These fields are always included in the output from AssignGeoTAXInfo.

Table 60: Match Codes

Response Element	Description
Ahh	Same as Shh, but indicates match to an alias name record or an alternate record.
Chh	Street address did not match, but located a street segment based on the input ZIP Code or city.
D00	Matched to a small town with P.O. Box or General Delivery only.
Ghh	Matched to an auxiliary file.
Hhh	House number was changed.
Qhh	Matched to USPS range records with unique ZIP Codes. CASS rules prohibit altering an input ZIP if it matches a unique ZIP Code value.
Rhh	Matched to a ranged address.
Shh	Matched to USPS data. This is considered the best address match, because it matched directly against the USPS list of addresses. S is returned for a small number of addresses when the matched address has a blank ZIP + 4.
Thh	Matched to a street segment record. Street segment records do not contain ZIP Code information. If you enter a ZIP Code, the application returns the ZIP Code you entered. If the input city and state has only one ZIP Code, the application returns that ZIP Code.
Uhh	Matched to USPS data but cannot resolve the ZIP + 4 code without the firm name or other information.
Xhhh	Matched to an intersection of two streets, for example, “Clay St & Michigan Ave.” The first hex digit refers to the last line information, the second hex digit refers to the first street in the intersection, and the third hex digit refers to the second street in the intersection. **Note:** The USPS does not allow intersections as a valid deliverable address
Yhhh	Same as Xhhh, but an alias name record was used for one or both streets.
Z	No address given, but verified the provided ZIP Code.

Hex digit decoding

The following table contains the description of the hex digits for the Match Code values.

Code	In first hex position means:	In second and third hex position means:
0	No change in last line.	No change in address line.
1	ZIP Code changed.	Street type changed.
2	City changed.	Pre-directional changed.
3	City and ZIP Code changed.	Street type and pre-directional changed.
4	State changed.	Post-directional changed.
5	State and ZIP Code changed.	Street type and post-directional changed.
6	State and City changed.	Pre-directional and post-directional changed.
7	State, City, and ZIP Code changed.	Street type, pre-directional, and postdirectional changed.
8	ZIP + 4 changed.	Street name changed.
9	ZIP and ZIP + 4 changed.	Street name and street type changed.
A	City and ZIP + 4 changed.	Street name and pre-directional changed.
B	City, ZIP, and ZIP + 4 changed.	Street name, street type, and pre-directional changed.
C	State and ZIP + 4 changed.	Street name and post-directional changed.
D	State, ZIP, and ZIP + 4 changed.	Street name, street type, and post-directional changed.
E	State, City, and ZIP + 4 changed.	Street name, pre-directional, and postdirectional changed.
F	State, City, ZIP, and ZIP + 4 changed.	Street name, street type, pre-directional, and post-directional changed.

Match Error Codes

The following table describes the values returned when an error occurs or the application cannot find a match.

Table 61: Result Indicator Match Error Codes

Response Element	Description
Ennn, where nnn =	Indicates an error, or no match. This can occur when the address entered does not exist in the database, or the address is badly formed and cannot be parsed correctly. The last three digits of an error code indicate which parts of an address the application could not match to the database.
000	No match made.
001	Low level error.
002	Could not find data file.
003	Incorrect GSD file signature or version ID.
010	No city and state or ZIP Code found.
011	Input ZIP not in the directory.

Response Element	Description
012	Input city not in the directory.
013	Input city not unique in the directory.
014	Out of licensed area. Only occurs if using Group 1 licensing technology.
015	Record count is depleted and license has expired.
020	No matching streets found in directory.
021	No matching cross streets for an intersection match.
022	No matching segments.
023	Unresolved match.
024	No matching segments. (Same as 022.)
025	Too many possible cross streets for intersection matching.
026	No address found when attempting a multiline match.
027	Invalid directional attempted.
028	Record also matched EWS data, therefore the application denied the match.
029	No matching range, single street segment found.
030	No matching range, multiple street segments found.

Location Codes

The Location Codes indicate the methodology used to compute the geocode and may also provide some information about the quality of the geocode.

A Location Code of 'E' indicates a location code is not available. This usually occurs when you have requested ZIP Code centroids of a high quality, and one is not available for that match. It can occur infrequently when the does not have a 5-digit centroid location. An 'E' location code type may also be returned when the input address cannot be standardized and there is no input ZIP Code. In this case, do not assume the ZIP Code returned with the nonstandardized address is the correct ZIP Code because the did not standardize the address; therefore, the does not return geocoding or Census Block information.

Address Location Codes

Address location codes detail the known qualities about the geocode. An address location code has the following characters.

**Characters	Description	
1st character	Always an 'A' indicating an address location.	
2nd character	May be one of the following:	
	C	Interpolated address point location.
	G	GeoTAX Auxiliary fiel or Landmark Auxiliary file data location.

**Characters	Description	
	I	Application infers the correct segment from the candidate records.
	P	Point-level data location.
	R	Location represents a ranged address.
	S	Location on a street range.
	X	Location on an intersection of two streets.
3rd and 4th characters	Digit indicating other qualities about the location.	

Table 62: Location Codes

**Response Element	Description
ACn - Interpolated address point location	
AIn - The correct segment is inferred from the candidate records at match time.	
ASn - House range address geocode. This is the most accurate street interpolated geocode available.	
where n =	
0	Best location.
1	Street side is unknown. The Census FIPS Block ID is assigned from the left side; however, there is no assigned offset and the point is placed directly on the street.
2	Indicates one or both of the following: • The address is interpolated onto a TIGER segment that did not initially contain address ranges. • The original segment name changed to match the USPS spelling. This specifically refers to street type, predirectional, and postdirectional. **Note:** Only the second case is valid for non-TIGER data because segment range interpolation is only completed for TIGER data.
3	Both 1 and 2.
7	Placeholder. Used when starting and ending points of segments contain the same value and shape data is not available.
AGn - Indicates a geocode match to a GeoTAX Auxiliary or Landmark Auxiliary file,	
where n =	
0	The geocode represents the center of a parcel or building.
1	The geocode is an interpolated address along a segment.
2	The geocode is an interpolated address along a segment, and the side of the street cannot be determined from the data provided in the auxiliary file record.
3	The geocode is the midpoint of the street segment.

**Response Element	Description
APnn - Indicates a point-level geocode match representing the center of a parcel or building, where nn =	
02	Parcel centroid Indicates the center of an assessor's parcel (tract or lot) polygon. When the center of an irregularly shaped parcel falls outside of its polygon, the centroid is manually repositioned to fall inside the polygon as closely as possible to the actual center.
04	Address point Represents field-collected GPS points with field-collected address data.
05	Structure centroid Indicates the center of a building footprint polygon, where the building receives mail or has telephone service. Usually a residential address consists of a single building. For houses with outbuildings (detached garages, shed, barns, etc.), only the residences have a structure point. Condominiums and duplexes have multiple points for each building. Larger buildings, such as apartment complexes, typically receive mail at one address for each building and therefore individual apartments are not represented as discrete structure points. Shopping malls, industrial complexes, and academic or medical center campuses where one building accepts mail for the entire complex are represented as one point. When addresses are assigned to multiple buildings within one complex, each addressed structure is represented by a point. If the center of a structure falls outside of its polygon, the center is manually repositioned to fall inside the polygon.
07	Manually placed Address points are manually placed to coincide with the midpoint of an assessor's parcel's street frontage at a distance from the center line.
08	Front door point Represents the designated primary entrance to a building. If a building has multiple entrances and there is no designated primary entrance or the primary entrance cannot readily be determined, the primary entrance is chosen based on proximity to the main access street and availability of parking.
09	Driveway offset point Represents a point located on the primary access road (most commonly a driveway) at a perpendicular distance of between 33- 98 feet (10-30 meters) from the main roadway.
10	Street access point Represents the primary point of access from the street network. This address point type is located where the driveway or other access road intersects the main roadway.
21	Base parcel point The Centrus point data includes individual parcels that may be "stacked". These stacked parcels are individually identified by their unit or suite number, and

**Response Element	Description
	the is able to match to this unit number and return the correct APN. If an input address is for a building or complex, without a unit number, the "base" parcel information returns and will not standardize to a unit number or return additional information such as an APN.
3	The geocode is the midpoint of the street segment.
ARn - Ranged address geocode,	
where n =	
1	The geocode is placed along a single street segment, midway between the interpolated location of the first and second input house numbers in the range.
2	The geocode is placed along a single street segment, midway between the interpolated location of the first and second input house numbers in the range, and the side of the street is unknown. The Census FIPS Block ID is assigned from the left side; however, there is no assigned offset and the point is placed directly on the street.
4	The input range spans multiple USPS segments. The geocode is placed on the endpoint of the segment which corresponds to the first input house number, closest to the end nearest the second input house number.
7	Placeholder. Used when the starting and ending points of the matched segment contain the same value and shape data is not available.
AXn - Intersection geocode,	
where n =	
3	Standard single-point intersection computed from the center lines of street segments.
8	Interpolated (divided-road) intersection geocode. Attempts to return a centroid for the intersection.

Street centroid location codes

Street centroid location codes indicate the Census ID accuracy and the position of the geocode on the returned street segment. A street centroid location code has the following characters.

Character	Description
1st character	Always 'C' indicating a location derived from a street segment.
2nd character	Census ID accuracy based on the search area used to obtain matching street segment.
3rd character	Location of geocode on the returned street segment.

The following table contains the values and descriptions for the street centroid location codes.

Character position	Code	Description
2nd Character		
	B	Block Group accuracy (most accurate). Based on input ZIP Code.

Character position	Code	Description
	T	Census Tract accuracy. Based on input ZIP Code.
	C	Unclassified Census accuracy. Normally accurate to at least the County level. Based on input ZIP Code.
	F	Unknown Census accuracy. Based on Finance area.
	P	Unknown Census accuracy. Based on input City.
3rd Character		
	C	Segment centroid.
	L	Segment low-range endpoint.
	H	Segment high-range endpoint.

ZIP + 4 Location Codes

ZIP + 4® centroid location codes indicate the quality of two location attributes: Census ID accuracy and positional accuracy. A ZIP + 4 centroid location code has the following characters.

Character	Description
1st character	Always a 'Z' indicating a location derived from a ZIP centroid..
2nd character	Census ID accuracy.
3rd character	Location type.
4th character	How the location and Census ID was defined. Provided for completeness, but may not be useful for most applications.

The following table contains the values and descriptions for the ZIP + 4 location codes.

Character position	Code	Description
2nd Character		
	B	Block Group accuracy (most accurate).
	T	Census Tract accuracy.
	C	Unclassified Census accuracy. Normally accurate to at least the County level. Based on input ZIP Code.
3rd Character		
	5	Location of the Post Office that delivers mail to the address, a 5-digit ZIP Code centroid, or a location based upon locale (city). See the 4th character for a precise indication of locational accuracy.
	7	Location based upon a ZIP + 2 centroid. These locations can represent a multiple block area in urban locations, or a slightly larger area in rural settings.
	9	Location based upon a ZIP + 4 centroid. These are the most accurate centroids and normally place the location on the correct block face. For a small number of records, the location may be the middle of the entire

Character position	Code	Description
		street on which the ZIP + 4 falls. See the 4th character for a precise indication of locational accuracy.
4th Character		
	A	Address matched to a single segment. Location assigned in the middle of the matched street segment, offset to the proper side of the street.
	a	Address matched to a single segment, but the correct side of the street is unknown. Location assigned in the middle of the matched street segment, offset to the left side of the street, as address ranges increase.
	B	Address matched to multiple segments, all segments have the same Block Group. Location assigned to the middle of the matched street segment with the most house number ranges within this ZIP + 4. Location offset to the proper side of the street.
	b	Same as methodology B except the correct side of the street is unknown. Location assigned in the middle of the matched street segment, offset to the left side of the street, as address ranges increase.
	C	Address matched to multiple segments, with all segments having the same Census Tract. Returns the Block Group representing the most households in this ZIP + 4. Location assigned to t he middle of the matched street segment with the most house number ranges within this ZIP + 4. Location offset to the proper side of the street.
	c	Same as methodology C except the correct side of the street is unknown. Location assigned in the middle of the matched street segment, offset to the left side of the street, as address ranges increase.
	D	Address matched to multiple segments, with all segments having the same County. Returns the Block Group representing the most households in this ZIP + 4. Location assigned to the middle of the matched street segment with the most house number ranges within this ZIP + 4. Location offset to the proper side of the street.
	d	Same as methodology D except the correct side of the street is unknown. Location assigned in the middle of the matched street segment, offset to the left side of the street, as address ranges increase.
	E	Street name matched; no house ranges available. All matched segments have the same Block Group. Location placed on the segment closest to the center of the matched segments. In most cases, this is on the mid-point of the entire street.
	F	Street name matched; no house ranges available. All matched segments have the same Census Tract. Location placed on the segment closest to the center

Character position	Code	Description
		of the matched segments. In most cases, this is on the mid-point of the entire street.
	G	Street name matched (no house ranges available). All matched segments have the same County. Location placed on the segment closest to the center of the matched segments. In most cases, this is on the mid-point of the entire street.
	H	Same as methodology G, but some segments are not in the same County. Used for less than .05% of the centroids.
	I	Created ZIP + 2 cluster centroid as defined by methodologies A, a, B, and b. All centroids in this ZIP + 2 cluster have the same Block Group. Location assigned to the ZIP + 2 centroid.
	J	Created ZIP + 2 cluster centroid as defined by methodologies A, a, B, b, C, and c. All centroids in this ZIP + 2 cluster have the same Census Tract. Location assigned to the ZIP + 2 centroid.
	K	Created ZIP + 2 cluster centroid as defined by methodologies A, a, B, b, C, c, D, and d. Location assigned to the ZIP + 2 centroid.
	L	Created ZIP + 2 cluster centroid as defined by methodology E. All centroids in this ZIP + 2 cluster have the same Block Group. Location assigned to the ZIP + 2 centroid.
	M	Created ZIP+2 cluster centroid as defined by methodology E and F. All centroids in this ZIP + 2 cluster have the same Census Tract. Location assigned to the ZIP + 2 centroid.
	N	Created ZIP + 2 cluster centroid as defined by methodology E, F, G, and H. Location assigned to the ZIP + 2 centroid.
	O	ZIP Code is obsolete and not currently used by the USPS. Historic location assigned.
	V	Over 95% of addresses in this ZIP Code are in a single Census Tract. Location assigned to the ZIP Code centroid.
	W	Over 80% of addresses in this ZIP Code are in a single Census Tract. Reasonable Census Tract accuracy. Location assigned to the ZIP Code centroid.
	X	Less than 80% of addresses in this ZIP Code are in a single Census Tract. Census ID is uncertain. Location assigned to the ZIP Code centroid.
	Y	Rural or sparsely populated area. Census code is uncertain. Location based upon the USGS places file.
	Z	P.O. Box or General Delivery addresses. Census code is uncertain. Location based upon the Post Office location that delivers the mail to that address.

Geographic Centroid Location Codes

Geographic centroid location codes indicate the quality of two location attributes: the geographic location and area type.

Character	Description
1st character	Always 'G' indicating a location derived from a geographic centroid.
2nd character	Geographic area type.

The following table contains the values and descriptions for the geographic centroid location codes.

Character position	Code	Description
2nd Character		
	M	Municipality (city).
	C	County.
	S	State.

GeoConfidence Module

GeoConfidenceSurface

GeoConfidenceSurface returns geoconfidence polygons (also called surfaces) based on the quality of the geocode information generated by the Enterprise Geocoding Module. With the geoconfidence polygons generated, you can then overlap this polygon with other spatial data to determine a risk or probability.

This service is used by the GeoConfidence Module's FloodZoneAnalysis dataflow template.

Note: GeoConfidence uses services provided by the Enterprise Geocoding and Location Intelligence modules.

Resource URL

JSON endpoint:

```
http://server:port/rest/GeoConfidenceSurface/results.json
```

XML endpoint:

```
http://server:port/rest/GeoConfidenceSurface/results.xml
```

Request

The input fields for GeoConfidenceSurface are the output fields returned by the GeoConfidence output category of the Enterprise Geocoding Module. These fields are described below.

Response Element	Max. Field Length (bytes)	Description
GeoConfidenceCode	13	The value returned in this field indicates which geoconfidence surface type has been returned. Possible values are: **INTERSECTION** A geocode point for the intersection of two streets. **ADDRESS** An array of street segment points representing the street segment where the address is located. **POINT** If the geocoder was able to match the address using point data, the point geometry where the address is located. **POSTAL1** A geocode point for the ZIP centroid. **POSTAL2** An array of points for all street segments in the ZIP + 2 in which the address is located. **POSTAL3** An array of points for street segments in the ZIP + 4 in which the address is located. **ERROR** An error has occurred.
StreetSegmentPoints	1024	An array of latitude/longitude values that represent the street segment points. **Note:** This field contains values only if the `GeoConfidenceCode` field returns a value of `ADDRESS`, `POSTAL2`, or `POSTAL3`.
GeoConfidenceCentroidLatitude	11	The latitude of the centroid of the geoconfidence polygon.
GeoConfidenceCentroidLongitude	12	The longitude of the centroid of the geoconfidence polygon.

Response

The GeoConfidenceSurface output field contains the geoconfidence polygon.

Response Element	Description
Geometry	A geoconfidence polygon that represents the returned geometry.

Global Sentry Module

GlobalSentry

The GlobalSentry service attempts to match transactions against government provided watch lists that contain data from various countries. The GlobalSentry service relies on the Universal Name, Data Normalization, and Advanced Matching modules. The service provides pre-configured and optimized rules for matching against the normalized and consolidated watch lists. These lists include the following:

- Denied Persons List (United States)
- Unverified List (BIS Red Flag) (United States)
- Consolidated Financial Sanction Targets (Individuals and Entities) (United Kingdom/European Union)
- Consolidated lists of persons, groups, and entities subject to EU financial sanctions (European Union)
- DFAT Consolidated List (Australia)
- OSFI Consolidated List (Individuals and Entities) (Canada)
- Specially Designated Nationals, Terrorists, Narcotic Traffickers and other Blocked Persons List (United States)
- Statutorily Debarred Parties List (United States)

Matches are performed against Sanctioned Countries, Name, Address, ID Number and other information such as DOB to provide an "Overall Risk Level Score" that allows your organization to make the right choice before making a decision to block a particular transaction and avoid false positive results.

The following steps describe how Global Sentry processes data:

1. The service first scans all required data in the transaction to identify countries that have been sanctioned. If a sanction country match has been identified, the transaction bypasses all other matching criteria and is assigned the highest possible risk score.
2. If a sanctioned country match has not been identified, the service then attempts to match the transaction against the Global Sentry database using the Global Sentry Name Check, Global Sentry Address Check or Global Sentry ID Number Check subflows.
3. The Global Sentry Name Check attempts to match individuals, entities and vessels. If a name match is identified a Name Score is returned from the service.
4. The Global Sentry Address Check attempts to match addresses within a country. If an Address match is identified an Address Score is returned from the service.
5. The Global Sentry ID Number Check attempts to match identification numbers such as Passport, National ID, SSN, Fiscal Code, and so on. If an ID Number match is identified an ID Number Score is returned from the service.
6. If a transaction is not identified as a Name, Address or ID Number match, the transaction record is written to the output and given an overall risk level score of zero.
7. If a transaction has been identified as a Name, Address or Identification Number match, the service attempts to match those transactions against the Global Sentry database using the Global Sentry Other Data Check subflow.
8. The Global Sentry Other Data Check attempts to match the Place of Birth, Date of Birth, Nationality or Citizenship. If a match is identified a Place of Birth Score, Date of Birth Score, Nationality Score or Citizenship Score is returned by the service.
9. Global Sentry assigns an Overall Risk Level score to each transaction. The score is a value between 0 and 16 and is returned in the OverallRiskLevel field. In calculating the risk level, Global Sentry takes into account what data was provided in the input record and which inputs, if any, matched entries in the Global Sentry database. Generally, a higher value indicates a higher risk associated with the transaction.

Resource URL

JSON endpoint:

```
http://server:port/rest/GlobalSentry/results.json
```

XML endpoint:

```
http://server:port/rest/GlobalSentry/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GlobalSentry/results.json?
Data.FirstName=Miguel&Data.LastName=Batista
```

The JSON returned by this request would be:

Note: Empty response elements have been removed from this example. Only the first response record shown.

```
{"Output": [
      {
      "OverallRiskLevel": "10",
      "SanctionedCountryIdentified": "No",
      "Status": "S",
      "FirstName": "Miguel",
      "LastName": "Batista",
      "PlaceOfBirth": "San Sebastian (Guipuzcoa)  Spain",
      "EntryID": "315",
      "InputFilteredFirstName": "Miguel",
      "InputFilteredLastName": "Batista",
      "InputFirstName": "Miguel",
      "InputLastName": "Batista",
      "ListType": "DFAT Consolidated List",
      "MatchKey1": "MGL",
      "MatchKey2": "BTST",
      "NameMatchIdentified": "Yes",
      "NameProvided": "Yes",
      "AddressProvided": "No",
      "IDNumberProvided": "No",
      "AddressMatchIdentified": "No",
      "IDNumberMatchIdentified": "No",
      "CitizenshipScore": "0",
      "CitizenshipMatchIdentified": "No",
      "CitizenshipUID": "",
      "DOBScore": "0",
      "DOBMatchIdentified": "No",
      "NationalityScore": "0",
      "NationalityMatchIdentified": "No",
      "PlaceOfBirthScore": "0",
      "PlaceOfBirthMatchIdentified": "No",
      "CitizenshipProvided": "No",
      "DOBProvided": "No",
      "NationalityProvided": "No",
      "PlaceOfBirthProvided": "No",
      "WatchListFirstName": "Miguel",
      "WatchListLastName": "ALBISU IRIARTE",
      "NameScore": "100",
      "user_fields": []
   }
]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GlobalSentry/results.xml?
Data.FirstName=Miguel&Data.LastName=Batista
```

The XML returned by this request would be:

Note: Empty response elements have been removed from this example. Only the first response record shown.

```
<xml.GlobalSentryResponse
xmlns="http://www.pb.com/spectrum/services/GlobalSentry">
   <Output>
      <Row>
         <OverallRiskLevel>10</OverallRiskLevel>
         <SanctionedCountryIdentified>No</SanctionedCountryIdentified>
         <Status>S</Status>
         <FirstName>Miguel</FirstName>
         <LastName>Batista</LastName>
         <PlaceOfBirth>San Sebastian (Guipuzcoa)  Spain</PlaceOfBirth>
         <EntryID>315</EntryID>
         <InputFilteredFirstName>Miguel</InputFilteredFirstName>
         <InputFilteredLastName>Batista</InputFilteredLastName>
         <InputFirstName>Miguel</InputFirstName>
         <InputLastName>Batista</InputLastName>
         <ListType>DFAT Consolidated List</ListType>
         <MatchKey1>MGL</MatchKey1>
         <MatchKey2>BTST</MatchKey2>
         <NameMatchIdentified>Yes</NameMatchIdentified>
         <NameProvided>Yes</NameProvided>
         <AddressProvided>No</AddressProvided>
         <IDNumberProvided>No</IDNumberProvided>
         <AddressMatchIdentified>No</AddressMatchIdentified>
         <IDNumberMatchIdentified>No</IDNumberMatchIdentified>
         <CitizenshipScore>0</CitizenshipScore>
         <CitizenshipMatchIdentified>No</CitizenshipMatchIdentified>
         <DOBScore>0</DOBScore>
         <DOBMatchIdentified>No</DOBMatchIdentified>
         <NationalityScore>0</NationalityScore>
         <NationalityMatchIdentified>No</NationalityMatchIdentified>
         <PlaceOfBirthScore>0</PlaceOfBirthScore>
         <PlaceOfBirthMatchIdentified>No</PlaceOfBirthMatchIdentified>
         <CitizenshipProvided>No</CitizenshipProvided>
         <DOBProvided>No</DOBProvided>
         <NationalityProvided>No</NationalityProvided>
         <PlaceOfBirthProvided>No</PlaceOfBirthProvided>
         <WatchListFirstName>Miguel</WatchListFirstName>
         <WatchListLastName>ALBISU IRIARTE</WatchListLastName>
         <NameScore>100</NameScore>
         <user_fields/>
      </Row>
   </Output>
</xml.GlobalSentryResponse>
```

Request

Parameters for Input Data

Table 63: Global Sentry Input Fields

Parameter	Description
Data.Name	Full name. Required if FirstName and LastName is not used.
Data.FirstName	First name or all name elements other than last name. Required if Name is not used.
Data.LastName	Last name only. Required if Name is not used.

Parameter	Description
Data.AddressLine1	The first address line. Recommended if provided.
Data.AddressLine2	The second address line. Recommended if provided.
Data.AddressLine3	The third address line. Recommended if provided.
Data.Country	Full country name. Required if address lines are used.
Data.IDNumber	Identification Number, such as SSN, Passport, Visa, and so on. Recommended if provided.
Data.PlaceOfBirth	Any place of birth data. Recommended if provided.
Data.DOB	Date Of Birth, in the format of Year, Month, Day. Recommended if provided.
Data.Citizenship	Full country name. Recommended if provided.
Data.Nationality	Full country name. Recommended if provided.

Response

Table 64: Global Sentry Service Output

Response Element	Description
Name	
InputName	Input Name from the original data source.
InputFilteredName	Input Name with titles, suffixes and special characters removed from the original data source.
Name	Name returned from database.
InputFirstName	Input First Name from the original data source.
InputFilteredFirstName	Input First Name with titles, suffixes and special characters removed from the original data source.
FirstName	First Name returned from database.
InputLastName	Input Last Name from the original data source.

Response Element	Description
InputFilteredLastName	Input Last Name with titles, suffixes and special characters removed from the original data source.
LastName	Last Name returned from database.
NameScore	Name match score. 0 - 100.
NameMatchIdentified	Identifies Name was a match. Yes or No
NameProvided	Identifies Name was provided in the input data . Yes or No
Address	
InputAddressLine1	Input Address line from the original data source.
AddressLine1	Address line returned from database.
InputAddressLine2	Input Address line from the original data source.
AddressLine2	Address line returned from database.
InputAddressLine3	Input Address line from the original data source.
AddressLine3	Address line returned from database.
AddressScore	Address match score. 0 - 100.
AddressMatchIdentified	Identifies Address was a match. Yes or No.
AddressProvided	Identifies Address was provided in the input data. Yes or No.
InputCountry	Input Country from the original data source.
Country	Country returned from database.
ID Number	
InputIDNumber	Input ID Number from the original data source.
IDNumber	ID Number returned from database.
IDNumberScore	ID Number match score. 0-100.
IDMatchNumberIdentified	Identifies ID Number was a match. Yes or No.
IDNumberProvided	Identifies ID Number was provided in the input data. Yes or No.
Place of Birth	
InputPlaceOfBirth	Input Place of Birth from the original data source.
PlaceOfBirth	Place of Birth returned from database.
PlaceOfBirthScore	Place of Birth match score. 0-100.
PlaceOfBirthMatchIdentified	Identifies Place of Birth was a match. Yes or No.
PlaceOfBirthProvided	Identifies Place of Birth was provided in the input data. Yes or No.
Date of Birth	
InputDOB	Input Date of Birth from the original data source.

Response Element	Description
DOB	Date of Birth returned from database.
DOBScore	Date of Birth match score. 0-100.
DOBMatchIdentified	Identifies Date of Birth was a match. Yes or No.
DOBProvided	Identifies Date of Birth was provided in the input data. Yes or No.
Citizenship	
InputCitizenship	Input Citizenship from the original data source.
Citizenship	Citizenship returned from database.
CitizenshipScore	Citizenship match score. 0 to 100.
CitizenshipMatchIdentified	Identifies Citizenship was a match. Yes or No.
CitizenshipProvided	Identifies Citizenship was provided in the input data. Yes or No.
Nationality	
InputNationality	Input Nationality from the original data source.
Nationality	Nationality returned from database.
NationalityScore	Nationality match score. 0-100.
NationalityMatchIdentified	Identifies Nationality was a match. Yes or No.
NationalityProvided	Identifies Nationality was provided in the input data. Yes or No.
Government List Information	
EntryID	Entry ID that identifies a name, entity, vessel, address, id number, place of birth, date of birth, citizenship or nationality. This is provided by each govt. agency.
ListType	Name of list provided by the government agencies. SDN, EU, Bank Of England, Financial Institutions of Canada.
Risk Analysis	
OverAllRiskLevel	Risk score per match. 0-16. For more information, see **Understanding the Risk Analysis Score** on page 149.
SanctionCountryIdentified	Sanction country was identified as a match. Yes or No.

Understanding the Risk Analysis Score

Risk analysis processing assigns a point value to each of these inputs depending on whether the input was provided and whether it matched a record in the Global Sentry database. The risk analysis score is the sum of these point values. Points are assigned as shown in the following table.

Table 65: Risk Analysis Scoring Method

Input	No Data Provided	Matched	Did Not Match
Name	0	4	0

Input	No Data Provided	Matched	Did Not Match
Address	1	2	0
ID	1	2	0
Date of Birth	1	2	0
Place of Birth	1	2	0
Citizenship	1	2	0
Nationality	1	2	0

Generally, each input that matches the database is assigned 2 points; Name is the exception. A name match scores 4 points. Name score is weighted higher following guidance from sources including OFAC, who indicate that a name match is more significant than other types of matches.

If an input is provided and does not match an entry on the database, it is assigned 0 points and has no effect on the overall risk level. This is consistent with guidance stating that a name match, coupled with a significant amount of additional data which does not match that entry in the database, should not be considered a "hit" against a particular list.

If an input is not provided, it is assigned a score of 1. This has the effect of identifying as higher risk those transactions where one or more inputs match the database, but there are some inputs which are not available for matching. For these types of transactions, the true risk level cannot be accurately calculated because of the missing data. Guidance from agencies such as OFAC suggests that in these cases you should attempt to obtain as much of the missing data as possible in order to return a more accurate assessment of the risk involved in the transaction.

Although higher scores indicate a higher risk transactions, the risk level alone is not always sufficient to determine the appropriate action. This is because different combinations of matched, not-matched, and not-provided inputs can result in the same score. To provide additional information to determine whether an interdiction is appropriate, the Global Sentry service also returns two indicators for each of the seven inputs that are used in matching. These indicate whether the input was provided and whether the input matched the database. This allows you to perform additional analysis on transactions that are in the middle of the risk spectrum to understand whether it is appropriate to report the transaction to the watch list authority, to flag the transaction as needing additional input data for an accurate risk assessment, to approve the transaction, or to take some other action.

Customizing the Global Sentry Service

Global Sentry deploys five dataflow templates that you can modify in Enterprise Designer. Each dataflow consists of various components that were installed from the Spectrum™ Technology Platform, Universal Name, Data Normalization and Advanced Matching modules.

The names of the dataflows are:

- Global Sentry
- Global Sentry Name Check
- Global Sentry Address Check
- Global Sentry ID Number Check
- Global Sentry Other Data Check
- Global Sentry Batch
- Global Sentry Name Check Batch
- Global Sentry Address Check Batch
- Global Sentry ID Number Check Batch
- Global Sentry Other Data Check Batch

Location Intelligence Module

Where to Find Documentation?

The Location Intelligence Module provides spatial services that allows you to determine relationships between locations, areas, or points of interest and other business data, and visually show these relationships on a map. These services include:

- Geometry
- Feature
- Mapping
- MapTiling
- Named Resource
- Web Feature Service
- Web Map Service
- Catalog Service for the Web

To learn about the Location Intelligence Module services, see the *Spectrum Spatial Guide* on **support.pb.com**.

Universal Addressing Module

AutoCompleteLoqate

AutoCompleteLoqate offers real-time entry of address data for fast, accurate results. Users are returned instant results based on each character entered into the form, ensuring only accurate data is entered into the database.

Resource URL

JSON endpoint:

```
http://server:port/rest/AutoCompleteLoqate/results.json
```

XML endpoint:

```
http://server:port/rest/AutoCompleteLoqate/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/AutoCompleteLoqate/results.json?Data.AddressLine1=1+Global
```

The JSON returned by this request would be:

Note: To make the example easier to read, empty response elements have been removed and only the first three address matches are shown.

```
{"output_port": [
      {
```

```
      "ProcessedBy": "LOQATE",
      "HouseNumber": "1",
      "AddressLine1": "1 Global Vw",
      "FirmName": "Map Info",
      "City": "Troy",
      "StateProvince": "NY",
      "PostalCode": "12180-8399",
      "Country": "United States",
      "PostalCode.AddOn": "8399",
      "user_fields": []
   },
      {
      "ProcessedBy": "LOQATE",
      "HouseNumber": "1",
      "AddressLine1": "1 Global Pl",
      "City": "Glendale",
      "StateProvince": "AZ",
      "PostalCode": "85306-3216",
      "Country": "United States",
      "PostalCode.AddOn": "3216",
      "user_fields": []
   },
      {
      "ProcessedBy": "LOQATE",
      "HouseNumber": "1",
      "AddressLine1": "1 Global Dr",
      "City": "Olive Hill",
      "StateProvince": "KY",
      "PostalCode": "41164-6739",
      "Country": "United States",
      "PostalCode.AddOn": "6739",
      "user_fields": []
   }
]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/AutoCompleteLoqate/results.xml?Data.AddressLine1=1+Global
```

The XML returned by this request would be:

Note: To make the example easier to read, empty response elements have been removed and only the first three address matches are shown.

```
<ns2:xml.AutoCompleteLoqateResponse
xmlns:ns2="http://www.pb.com/spectrum/services/AutoCompleteLoqate">
   <ns2:output_port>
      <ns2:Address>
         <ns2:ProcessedBy>LOQATE</ns2:ProcessedBy>
         <ns2:HouseNumber>1</ns2:HouseNumber>
         <ns2:AddressLine1>1 Global Vw</ns2:AddressLine1>
         <ns2:FirmName>Map Info</ns2:FirmName>
         <ns2:City>Troy</ns2:City>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>12180-8399</ns2:PostalCode>
         <ns2:PostalCode.AddOn>8399</ns2:PostalCode.AddOn>
         <ns2:Country>United States</ns2:Country>
      </ns2:Address>
      <ns2:Address>
         <ns2:ProcessedBy>LOQATE</ns2:ProcessedBy>
         <ns2:HouseNumber>1</ns2:HouseNumber>
         <ns2:AddressLine1>1 Global Pl</ns2:AddressLine1>
         <ns2:City>Glendale</ns2:City>
         <ns2:StateProvince>AZ</ns2:StateProvince>
         <ns2:PostalCode>85306-3216</ns2:PostalCode>
         <ns2:PostalCode.AddOn>3216</ns2:PostalCode.AddOn>
         <ns2:Country>United States</ns2:Country>
      </ns2:Address>
      <ns2:Address>
         <ns2:ProcessedBy>LOQATE</ns2:ProcessedBy>
         <ns2:HouseNumber>1</ns2:HouseNumber>
         <ns2:AddressLine1>1 Global Dr</ns2:AddressLine1>
         <ns2:City>Olive Hill</ns2:City>
         <ns2:StateProvince>KY</ns2:StateProvince>
         <ns2:PostalCode>41164-6739</ns2:PostalCode>
         <ns2:PostalCode.AddOn>6739</ns2:PostalCode.AddOn>
         <ns2:Country>United States</ns2:Country>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.AutoCompleteLoqateResponse>
```

Request

Parameters for Input Data

The following table lists the input for AutoCompleteLoqate.

Table 66: Input Format

Parameter	Description
Data.AddressLine1	The first address line.
Data.AddressLine2	The second address line.
Data.AddressLine3	The third address line.
Data.AddressLine4	The fourth address line.
Data.City	The city name.
Data.Country	The country code or name, in any of the following formats: • 2-digit ISO country code • 3-digit UPU Country code • English country name For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Data.FirmName	The company or firm name.
Data.PostalCode	The postal code for the address.
Data.StateProvince	The state or province.

Parameters for Options

Table 67: AutoCompleteLoqate Options

Parameter	Description
Option.Database.Loqate	Specifies the database to be used for address processing. Only databases that have been defined in the **Database Resources** panel in the Management Console are available.

Parameter	Description
Option.OutputCasing	Specifies the casing of the output data. One of the following: **M** The output in mixed case (default). For example: 123 Main St Mytown FL 12345 **U** The output in upper case. For example: 123 MAIN ST MYTOWN FL 12345
Option.HomeCountry	Specifies the default country. You should specify the country where most of your addresses reside. For example, if most of the addresses you process are in Canada, specify Canada. The valid country names are: Afghanistan, Albania, Algeria, American Somoa, Andorra, Angola, Anguilla, Antigua And Barbuda, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, Bosnia And Herzegovina, Botswana, Brazil, British Virgin Islands, Brunei Darussalam, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Cayman Islands, Central African Republic, Chad, Chile, China, Colombia, Comoros Islands, Congo, Cook Islands, Costa Rica, Cote D'Ivoire, Croatia, Cuba, Cyprus, Czech Republic, Democratic Republic Of Congo, Denmark, Djibouti, Dominica, Dominican Republic, East Timor, Ecuador, Egypt, El Salvador, Equitorial Guinea, Eritrea, Estonia, Ethiopia, Falkland Islands, Faroe Islands, Federated States Of Micronesia, Fiji, Finland, France, French Guiana, Gabon, Gambia, Germany, Ghana, Gibraltar, Greece, Greenland, Grenada, Guadeloupe, Guam, Guatemala, Guinea, Guinea Bissau, Guyana, Haiti, Holy See, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, Israel, Italy, Jamaica, Japan, Jordan, Kazakhstan, Kenya, Kiribati, Korea, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Marshall Islands, Martinique, Mauritania, Mauritius, Mayotte, Mexico, Moldova, Monaco, Mongolia, Monserrat, Morocco, Mozambique, Myanmar, Namibia, Nauru, Nepal, Netherlands Antilles, New Caledonia, New Zealand, Nicaragua, Niger, Nigeria, Niue, Norway, Oman, Pakistan, Palau, Panama, Papua New Guinea, Paraguay, Peru, Philippines, Pitcairn Islands, Poland, Portugal, Puerto Rico, Qatar, Republic Of Georgia, Republic Of Korea, Republic Of Singapore, Reunion, Romania, Russia, Rwanda, Saint Helena, Saint Kitts And Nevis, Saint Lucia, Saint Pierre And Miquelon, Saint Vincent And The Grenadines, Samoa, San Marino, Sao Tome And Principe, Saudi Arabia, Senegal, Seychelles, Sierra Leone, Slovakia, Slovenia, Solomon Islands, Somalia, South Africa, Spain, Sri Lanka, Sudan, Surivalue, Swaziland, Sweden, Switzerland, Syria, Tahiti, Taiwan, Tajikistan, Tanzania, Thailand, The Netherlands, Togo, Tonga, Trinidad And Tobago, Tristan Da Cunha, Tunisia, Turkey, Turkmenistan, Turks And Caicos Islands, Tuvalu, Uganda, Ukraine, United Arab Emirates, United Kingdom, United States, Uruguay, Uzbekistan, Vanuatu, Venezuela, Vietnam, Virgin Islands (US), Wallis And Futuna, Yemen, Yugoslavia, Zambia, Zimbabwe

Parameter	Description
Option.OutputCountryFormat	Specifies the format to use for the country name returned in the **Country** output field. For example, if you select English, the country name "Deutschland" would be returned as "Germany". **E** Use English country names (default). **I** Use two-letter ISO abbreviation for the countries instead of country names. **U** Use Universal Postal Union abbreviation for the countries instead of country names.
Option.OutputScript	Specifies the alphabet or script in which the output should be returned. This option is bi-directional and generally takes place from Native to Latin and Latin to Native. **Input** Do not perform transliteration and provide output in the same script as the input (default). **Native** Output in the native script for the selected country wherever possible. **Latn** Use English values.
Option.isPowersearchEnable	Reduces input time by up to 80% for 240 countries by using data in the form of an index file. When you conduct a search, the Loqate Engine will first look for the corresponding index. If present, the method will attempt to instantly return a list of candidate addresses. If the index is not present, or if the index does not return any results, the original search process will be triggered. **Note:** Powersearch can be performed when there are two and only two fields in the input file: the Country field and any one of the AddressLine fields. If you select this option and your input file contains additional fields, the original search process will automatically be triggered. To conduct its search, Auto Complete indexes use up to the first 10 characters for searches within the United States and up to the first 15 characters for searches within all other eligible countries. Spaces and punctuation are not factored into this count. Powersearch cannot be used for the following countries: Botswana, Ethiopia, India, Kazakhstan, Malaysia, Mongolia, Saint Kitts and Nevis, and San Marino. **Note:** You must have a valid license for Powersearch processing. If you select this option but are not licensed for Powersearch, or if your license has expired, you will receive an error.
Option.MaximumResults	The maximum number of addresses that AutoCompleteLoqate should return. The default is 10.
Option.FailJobOnDataLicenseError	Specifies how you want Spectrum Technology Platform to respond when a data license error occurs. **Fail the job** Fail the entire job if a data license error occurs. **Fail the record** Fail the record(s) for which the data license error occurs and continue processing.

Response

The output from AutoCompleteLoqate is optional and corresponds directly to the fields you selected in the Output Fields section of the AutoCompleteLoqate Options dialog box.

Table 68: AutoCompleteLoqate Output

Response Element	Description
AddressLine1	The first address line.
AddressLine2	The second address line.
AddressLine3	The third address line.
AddressLine4	The fourth address line.
City	The city name.
Country	The three-character ISO 3116-1 Alpha-3 code for the country. For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
FirmName	The firm name.
HouseNumber	The ending house number for the range in which the candidate address's house number falls.
PostalCode	The postal code.
PostalCode.AddOn	The last four digits of the ZIP + 4® Code.
ProcessedBy	Indicates which address coder processed the address. **LOQATE** — The Loqate coder processed the address.
StateProvince	The state or province abbreviation.
Status	Reports the success or failure of the match attempt. **null** — Success **F** — Failure
Status.Code	The reason for failure, if there is one. • DisabledCoder • RequestFailed • NoLookupAddressFound
Status.Description	A description of the problem, if there is one. **Did not return multiples** — The input address matched only one address in the database. AutoCompleteLoqate returns data only if multiple possible matches were found. **Not able to look up the address pattern** — AutoCompleteLoqate is not able to process the partial address.

AutoCompleteLoqate Sample Web Application

You can access a sample web application that demonstrates the Auto Complete Loqate functionality. When you enter a partial address, this application makes a call to the Auto Complete Loqate REST web service, which returns a suggested address.

Note: Prior to using this feature, you must add an Auto Complete Loqate database resource in Management Console and save the database resource in the Auto Complete Loqate Service.

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser and go to: `http://<servername>:<port>/autocomplete`. For example, if your server is named "myserver" and it uses the default HTTP port 8080, you would go to: `http://myserver:8080/autocomplete`.

   Note: This site is best viewed in Internet Explorer 8.0 or later, Chrome, or Mozilla Firefox.

3. When the login screen appears, enter "**guest**" as the user name and leave the password field blank.
4. Press **OK**.
5. Select a country from the drop-down list.
6. Begin typing your address in any of the fields provided.
7. Select from the list of suggested addresses.
8. To begin a new call, click **Reset**, which will clear the fields you used in your previous call.

GetCandidateAddresses

GetCandidateAddresses returns a list of addresses that are considered matches for a given input address. GetCandidateAddresses returns candidate addresses only if the input address matches multiple addresses in the postal database. If the input address matches only one address in the postal database, then no address data is returned.

For addresses outside the U.S. and Canada, you may notice inconsistent results between the multiple matches returned by ValidateAddress and the results for that same address returned by GetCandidateAddresses. If you experience inconsistent results, it is likely because you set the performance tuning setting in ValidateAddress to a value other than 100. To obtain consistent results between GetCandidateAddresses and ValidateAddress, set the performance tuning option to 100.

Note: By default, GetCandidateAddresses does not match to individual house numbers. Rather, it uses house number ranges for each street. After GetCandidateAddresses has determined the street name, city name, state/province name, and postal code, it checks to make sure the input house number falls within one of the ranges of house numbers given for the matched street name. The same type of logic applies to unit numbers. If you want to determine that an individual house number is valid, you should use the ValidateAddress Delivery Point Validation (DPV) processing option. DPV processing is only available for U.S. addresses.

The Canadian coder contains a reverse lookup routine that takes as input a specific postal code and returns the street information stored in the database for that postal code. To use this function enter nothing but a Canadian postal code in the PostalCode field. See the second example to view the return from a sample postal code.

GetCandidateAddresses is part of the Universal Addressing Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/GetCandidateAddresses/results.json
```

XML endpoint:

```
http://server:port/rest/GetCandidateAddresses/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GetCandidateAddresses/results.json?
Data.AddressLine1=P.O.+Box+1&Data.City=New+York&Data.StateProvince=NY
```

The JSON returned by this request would be:

```
{"output_port": [
      {
      "ProcessedBy": "USA",
      "RecordType": "PostOfficeBox",
      "MatchLevel": "A",
      "AddressLine1": "PO Box 1",
      "HouseNumberLow": "1",
      "HouseNumberHigh": "60",
      "HouseNumberParity": "B",
      "UnitNumberLow": "",
      "UnitNumberHigh": "",
      "UnitNumberParity": " ",
      "FirmName": "",
      "City": "New York",
      "USUrbanName": "",
      "StateProvince": "NY",
      "PostalCode": "10002",
      "Country": "USA",
      "PostalCode.AddOn": "0001",
      "user_fields": []
   },
      {
      "ProcessedBy": "USA",
      "RecordType": "PostOfficeBox",
      "MatchLevel": "A",
      "AddressLine1": "PO Box 1",
      "HouseNumberLow": "1",
      "HouseNumberHigh": "9",
      "HouseNumberParity": "B",
      "UnitNumberLow": "",
      "UnitNumberHigh": "",
      "UnitNumberParity": " ",
      "FirmName": "",
      "City": "New York",
      "USUrbanName": "",
      "StateProvince": "NY",
      "PostalCode": "10008",
      "Country": "USA",
      "PostalCode.AddOn": "0001",
      "user_fields": []
   },
      {
      "ProcessedBy": "USA",
      "RecordType": "PostOfficeBox",
      "MatchLevel": "A",
      "AddressLine1": "PO Box 1",
      "HouseNumberLow": "1",
      "HouseNumberHigh": "60",
      "HouseNumberParity": "B",
      "UnitNumberLow": "",
      "UnitNumberHigh": "",
      "UnitNumberParity": " ",
      "FirmName": "",
      "City": "New York",
      "USUrbanName": "",
      "StateProvince": "NY",
      "PostalCode": "10009",
      "Country": "USA",
      "PostalCode.AddOn": "0001",
      "user_fields": []
   }
]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GetCandidateAddresses/results.xml?
Data.AddressLine1=P.O.+Box+1&Data.City=New+York&Data.StateProvince=NY
```

The XML returned by this request would be:

```
<ns2:xml.GetCandidateAddressesResponse
xmlns:ns2="http://www.pb.com/spectrum/services/GetCandidateAddresses">
   <ns2:output_port>
      <ns2:Address>
         <ns2:ProcessedBy>USA</ns2:ProcessedBy>
         <ns2:RecordType>PostOfficeBox</ns2:RecordType>
         <ns2:MatchLevel>A</ns2:MatchLevel>
         <ns2:AddressLine1>PO Box 1</ns2:AddressLine1>
         <ns2:HouseNumberLow>1</ns2:HouseNumberLow>
         <ns2:HouseNumberHigh>60</ns2:HouseNumberHigh>
         <ns2:HouseNumberParity>B</ns2:HouseNumberParity>
         <ns2:UnitNumberLow/>
         <ns2:UnitNumberHigh/>
         <ns2:UnitNumberParity></ns2:UnitNumberParity>
         <ns2:FirmName/>
         <ns2:City>New York</ns2:City>
         <ns2:USUrbanName/>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>10002</ns2:PostalCode>
         <ns2:PostalCode.AddOn>0001</ns2:PostalCode.AddOn>
         <ns2:Country>USA</ns2:Country>
         <ns2:user_fields/>
      </ns2:Address>
      <ns2:Address>
         <ns2:ProcessedBy>USA</ns2:ProcessedBy>
         <ns2:RecordType>PostOfficeBox</ns2:RecordType>
         <ns2:MatchLevel>A</ns2:MatchLevel>
         <ns2:AddressLine1>PO Box 1</ns2:AddressLine1>
         <ns2:HouseNumberLow>1</ns2:HouseNumberLow>
         <ns2:HouseNumberHigh>9</ns2:HouseNumberHigh>
         <ns2:HouseNumberParity>B</ns2:HouseNumberParity>
         <ns2:UnitNumberLow/>
         <ns2:UnitNumberHigh/>
         <ns2:UnitNumberParity></ns2:UnitNumberParity>
         <ns2:FirmName/>
         <ns2:City>New York</ns2:City>
         <ns2:USUrbanName/>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>10008</ns2:PostalCode>
         <ns2:PostalCode.AddOn>0001</ns2:PostalCode.AddOn>
         <ns2:Country>USA</ns2:Country>
         <ns2:user_fields/>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.GetCandidateAddressesResponse>
```

Request

Parameters for Input Data

The following table lists the input for GetCandidateAddresses.

Table 69: Input Format

Parameter	Description
Data.AddressLine1	The first address line.
Data.AddressLine2	The second address line.
Data.AddressLine3	The third address line. Does not apply to U.S. and Canadian addresses.
Data.AddressLine4	The fourth address line. Does not apply to U.S. and Canadian addresses.
Data.AddressLine5	The fifth address line. Applies only to U.K. addresses. May contain street name, unit number, building number, and so on.
Data.City	The city name.
Data.StateProvince	The state or province. For U.S. addresses only, you may put the state in the City field instead of the StateProvince field.
Data.PostalCode	The postal code for the address. For U.S. addresses this is the ZIP Code™ in one of the following formats: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999 **Note:** For Canadian addresses you can complete just this field and have candidate address data returned. For other countries, AddressLine1 and AddressLine2 must also be completed.
Data.Country	The country code or name, in any of the following formats: • 2-digit ISO country code • 3-digit UPU Country code • English country name • French country name • German country name • Spanish country name For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Data.FirmName	The company or firm name.
Data.USUrbanName	U.S. address urbanization name. Used primarily for Puerto Rico addresses.

Parameters for Options

Table 70: GetCandidateAddresses Options

Parameter	Description
Option.PerformUSProcessing	Specifies whether or not to process U.S. addresses. If you enable U.S. address processing GetCandidateAddresses will attempt to retrieve candidate addresses for U.S. addresses. If you disable U.S. address processing, U.S. addresses will fail, meaning they are returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for U.S. address processing you must disable U.S. address processing in order for your jobs to complete successfully, regardless of whether or not they contain U.S. addresses. **Note:** You must have a valid license for U.S. address processing to successfully process U.S. addresses. If you enable U.S. address processing but are not licensed for this feature, or your license has expired, you will receive an error. **Y** Yes, process U.S. addresses (default). **N** No, do not process U.S. addresses.
Option.Database.US	Specifies the database to be used for U.S. address processing. Only databases that have been defined in the **US Database Resources** panel in the Management Console are available.
Option.PerformCanadianProcessing	Specifies whether or not to process Canadian addresses. If you enable Canadian address processing GetCandidateAddresses will attempt to retrieve candidate addresses for Canadian addresses. If you disable Canadian address processing, Canadian addresses will fail, meaning they are returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for Canadian address processing you must disable Canadian address processing in order for your jobs to complete successfully, regardless of whether or not they contain Canadian addresses. **Note:** You must have a valid license for Canadian address processing to successfully process Canadian addresses. If you enable Canadian address processing but are not licensed for this feature, or your license has expired, you will receive an error. **Y** Yes, process Canadian addresses (default). **N** No, do not process Canadian addresses.
Option.Database.Canada	Specifies the database to be used for Canadian address processing. Only databases that have been defined in the **Canadian Database Resources** panel in the Management Console are available.

Parameter	Description
Option.PerformInternationalProcessing	Specifies whether or not to process international addresses (addresses outside the U.S. and Canada). If you enable international address processing GetCandidateAddresses will attempt to retrieve candidate addresses for international addresses. If you disable international address processing, international addresses will fail, meaning they are returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for international address processing you must disable international address processing in order for your jobs to complete successfully, regardless of whether or not they contain international addresses. **Note:** You must have a valid license for international address processing to successfully process international addresses. If you enable international address processing but are not licensed for this feature, or your license has expired, you will receive an error. **Y** Yes, process international addresses (default). **N** No, do not process international addresses.
Option.Database.International	Specifies the database to be used for international address processing. Only databases that have been defined in the **International Database Resources** panel in the Management Console are available.
Option.OutputCasing	Specifies the casing of the output data. One of the following: **M** The output is in mixed case (default). For example: 123 Main St Mytown FL 12345 **U** The output is in upper case. For example: 123 MAIN ST MYTOWN FL 12345
Option.MaximumResults	The maximum number of candidate addresses that GetCandidateAddresses should return. The default is 10. The maximum is 10.
Option.OutputShortCityName	For U.S. addresses, specifies whether or not to return the USPS®-approved abbreviation for the city, if there is one. The USPS® provides abbreviations for city names that are 14 characters long or longer. City abbreviations are 13 characters or less and can be used when there is limited space on the mailing label. If there is no short city name for the city, then the full city name is returned. **Y** Yes, return the short city name. **N** No, do not return the short city name.

Parameter	Description
Option.DualAddressLogic	(U.S. addresses only). Controls whether GetCandidateAddresses should return a street match or a PO Box/Rural Route/Highway Contract match when the address contains both street and PO Box/Rural Route/Highway Contract information. For more information, see **About Dual Address Logic** on page 193. **N** (Default) USPS® CASS™ regulations determine the address returned based on the following order of priority: **1.** PO Box **2.** Firm **3.** Highrise **4.** Street **5.** Rural Route **6.** General Delivery **S** Return a street match, regardless of the address line. **P** Return a PO Box match, regardless of the address line.
Option.StreetMatchingStrictness	The strictness of the street name match (U.S. addresses only). **E** The input street name must match the database exactly. **T** The matching algorithm is "tight." **M** The matching algorithm is "medium" (default). **L** The matching algorithm is "loose".
Option.FirmMatchingStrictness	The strictness of the firm name match (U.S. addresses only). **E** The input firm name must match the database exactly. **T** The matching algorithm is "tight." **M** The matching algorithm is "medium" (default). **L** The matching algorithm is "loose."
Option.DirectionalMatchingStrictness	The strictness of the directional match. **E** The input directional must match the database exactly. **T** The matching algorithm is "tight." **M** The matching algorithm is "medium" (default). **L** The matching algorithm is "loose."
Option.PerformESM	Specifies whether or not to perform Enhanced Street Matching (ESM). ESM applies extra matching logic with additional data to any input address that is not matched through the regular address validation process. ESM applies to U.S. addresses only.

Parameter	Description
	Y Yes, perform ESM processing. **N** No, do not perform ESM processing (default).
Option.AddressLineSearchOnFail	Specifies whether ValidateAddress will search address lines for the city, state/province, and postal code. This option enables ValidateAddress to search the AddressLine input fields for the city, state/province, postal code, and country when the address cannot be matched using the values in the City, StateProvince, and PostalCode input fields. Consider enabling this option if your input addresses have the city, state/province, and postal code information in the AddressLine fields. Consider disabling this option if your input addresses use the City, State/Province and PostalCode fields. If you enable this option and these fields are used, there is an increased possibility that ValidateAddress will fail to correct values in these fields (for example a misspelled city name). **Y** Yes, search the address line fields (default). **N** No, do not search the AddressLine fields.

Response

GetCandidateAddresses returns the following output.

Table 71: GetCandidateAddresses Output

Response Element	Description
AddressLine1	The first address line.
AddressLine2	The second address line.
AddressLine3	The third address line.
AddressLine4	The fourth address line.
AddressLine5	For U.K. addresses only. If the address was validated, the fifth line of the validated and standardized address. If the address could not be validated, the fifth line of the input address without any changes.
City	The city name.
Country	The three-character ISO 3116-1 Alpha-3 code for the country. For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
FirmName	The firm name.
HouseNumberHigh	The ending house number for the range in which the candidate address's house number falls.

Response Element	Description
HouseNumberLow	The beginning house number for the range in which the candidate address's house number falls.
HouseNumberParity	Indicates the numbering scheme for the house numbers between HouseNumberLow and HouseNumberHigh, as follows: **E** Only even values **O** Only odd values **B** Both
MatchLevel	For addresses outside the U.S. and Canada, identifies the match level for the candidate address. U.S. and Canadian addresses are always "A." One of the following: **A** The candidate matches the input address at the street level. **B** The candidate matches the input address at the state/province level.
PostalCode	The postal code. In the U.S. this is the ZIP Code™.
PostalCode.AddOn	The last four digits of the ZIP + 4® Code. U.S. addresses only.
RecordType	The type of address record, as defined by U.S. and Canadian postal authorities (U.S. and Canadian addresses only): • FirmRecord • GeneralDelivery • HighRise • PostOfficeBox • RRHighwayContract • Normal
RecordType.Default	Code indicating the "default" match: **Y** The address matches a default record. **null** The address does not match a default record.
StateProvince	The state or province abbreviation.
Status	Reports the success or failure of the match attempt. **null** Success **F** Failure
Status.Code	The reason for failure, if there is one. There is only one possible value: • DisabledCoder • RequestFailed
Status.Description	A description of the problem, if there is one. **Did not return multiples** The input address matched only one address in the database. GetCandidateAddresses only returns data if multiple possible matches were found.

Response Element	Description
	Number of candidates is not greater than 1 The input address matched more than one address in the database but no addresses were returned. **PerformUSProcessing disabled** This value will appear if Status.Code=DisabledCoder. **PerformCanadianProcessing disabled** This value will appear if Status.Code=DisabledCoder. **PerformInternationalProcessing disabled** This value will appear if Status.Code=DisabledCoder.
UnitNumberHigh	The ending unit number for the range in which the candidate address's unit number falls.
UnitNumberLow	The beginning unit number for the range in which the candidate address's unit number falls.
UnitNumberParity	Indicates the numbering scheme for the unit numbers between UnitNumberLow and UnitNumberHigh, as follows: **E** Only even values **O** Only odd values **B** Both
USUrbanName	The validated city urbanization name. Urbanization names are used primarily for Puerto Rico addresses.

GetCandidateAddressesLoqate

GetCandidateAddressesLoqate returns a list of addresses that are considered matches for a given input address. GetCandidateAddressesLoqate returns candidate addresses only if the input address matches multiple addresses in the postal database. If the input address matches only one address in the postal database, then no address data is returned. The Country input field is required; if this field is blank, no output will be returned.

Note: By default, GetCandidateAddressesLoqate does not match to individual house numbers. Rather, it uses house number ranges for each street. After GetCandidateAddressesLoqate has determined the street name, city name, state/province name, and postal code, it checks to make sure the input house number falls within one of the ranges of house numbers given for the matched street name. The same type of logic applies to unit numbers.

GetCandidateAddressesLoqate is part of the Universal Addressing Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/GetCandidateAddressesLoqate/results.json
```

XML endpoint:

```
http://server:port/rest/GetCandidateAddressesLoqate/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GetCandidateAddressesLoqate/results.json?
Data.AddressLine1=PO+Box+1&Data.City=New+York&Data.StateProvince=NY
```

The JSON returned by this request would be:

Note: Empty response elements have been removed from this example. Only the first two candidate address are shown.

```
{"output_port": [
      {
      "ProcessedBy": "LOQATE",
      "AddressLine1": "PO Box 101",
      "City": "New York Mls",
      "StateProvince": "NY",
      "PostalCode": "13417-0101",
      "Country": "USA",
      "PostalCode.AddOn": "0101",
      "user_fields": []
   },
      {
      "ProcessedBy": "LOQATE",
      "AddressLine1": "PO Box 102",
      "City": "New York Mls",
      "StateProvince": "NY",
      "PostalCode": "13417-0102",
      "Country": "USA",
      "PostalCode.AddOn": "0102",
      "user_fields": []
   }
]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GetCandidateAddressesLoqate/results.xml?
Data.AddressLine1=PO+Box+1&Data.City=New+York&Data.StateProvince=NY
```

The XML returned by this request would be:

Note: Empty response elements have been removed from this example. Only the first two candidate address are shown.

```
<ns2:xml.GetCandidateAddressesLoqateResponse
xmlns:ns2="http://www.pb.com/spectrum/services/GetCandidateAddressesLoqate">

   <ns2:output_port>
      <ns2:Address>
         <ns2:ProcessedBy>LOQATE</ns2:ProcessedBy>
         <ns2:AddressLine1>PO Box 101</ns2:AddressLine1>
         <ns2:City>New York Mls</ns2:City>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>13417-0101</ns2:PostalCode>
         <ns2:PostalCode.AddOn>0101</ns2:PostalCode.AddOn>
         <ns2:Country>USA</ns2:Country>
      </ns2:Address>
      <ns2:Address>
         <ns2:ProcessedBy>LOQATE</ns2:ProcessedBy>
         <ns2:AddressLine1>PO Box 102</ns2:AddressLine1>
         <ns2:City>New York Mls</ns2:City>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>13417-0102</ns2:PostalCode>
         <ns2:PostalCode.AddOn>0102</ns2:PostalCode.AddOn>
         <ns2:Country>USA</ns2:Country>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.GetCandidateAddressesLoqateResponse>
```

Request

Parameters for Input Data

The following table lists the input for GetCandidateAddressesLoqate.

Table 72: Input Format

Parameter	Description
Data.AddressLine1	The first address line.
Data.AddressLine2	The second address line.
Data.AddressLine3	The third address line.
Data.AddressLine4	The fourth address line.
Data.City	The city name.
Data.Country	The country code or name, in any of the following formats: • 2-digit ISO country code • 3-digit UPU Country code • English country name For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292. **Note:** This field is required. If this field is blank, no output will be returned.
Data.FirmName	The company or firm name.
Data.PostalCode	The postal code for the address. For U.S. addresses this is the ZIP Code™ in one of the following formats:
Data.StateProvince	The state or province. For U.S. addresses only, you may put the state in the City field instead of the StateProvince field.

Parameters for Options

Table 73: GetCandidateAddressesLoqate Options

Parameter	Description
Option.Database.Loqate	Specifies the database to be used for address processing. Only databases that have been defined in the Management Console are available.
Option.OutputCasing	Specifies the casing of the output data. One of the following:

Parameter	Description
	M The output is in mixed case (default). For example: 123 Main St Mytown FL 12345 **U** The output is in upper case. For example: 123 MAIN ST MYTOWN FL 12345
Option.HomeCountry	Specifies the default country. You should specify the country where most of your addresses reside. For example, if most of the addresses you process are in Canada, specify Canada. GetCandidateAddressLoqate uses the country you specify to attempt validation when it cannot determine the country from the StateProvince, PostalCode, and Country address fields. The valid country names are: Afghanistan, Albania, Algeria, American Somoa, Andorra, Angola, Anguilla, Antigua And Barbuda, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, Bosnia And Herzegovina, Botswana, Brazil, British Virgin Islands, Brunei Darussalam, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Cayman Islands, Central African Republic, Chad, Chile, China, Colombia, Comoros Islands, Congo, Cook Islands, Costa Rica, Cote D'Ivoire, Croatia, Cuba, Cyprus, Czech Republic, Democratic Republic Of Congo, Denmark, Djibouti, Dominica, Dominican Republic, East Timor, Ecuador, Egypt, El Salvador, Equitorial Guinea, Eritrea, Estonia, Ethiopia, Falkland Islands, Faroe Islands, Federated States Of Micronesia, Fiji, Finland, France, French Guiana, Gabon, Gambia, Germany, Ghana, Gibraltar, Greece, Greenland, Grenada, Guadeloupe, Guam, Guatemala, Guinea, Guinea Bissau, Guyana, Haiti, Holy See, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, Israel, Italy, Jamaica, Japan, Jordan, Kazakhstan, Kenya, Kiribati, Korea, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Marshall Islands, Martinique, Mauritania, Mauritius, Mayotte, Mexico, Moldova, Monaco, Mongolia, Monserrat, Morocco, Mozambique, Myanmar, Namibia, Nauru, Nepal, Netherlands Antilles, New Caledonia, New Zealand, Nicaragua, Niger, Nigeria, Niue, Norway, Oman, Pakistan, Palau, Panama, Papua New Guinea, Paraguay, Peru, Philippines, Pitcairn Islands, Poland, Portugal, Puerto Rico, Qatar, Republic Of Georgia, Republic Of Korea, Republic Of Singapore, Reunion, Romania, Russia, Rwanda, Saint Helena, Saint Kitts And Nevis, Saint Lucia, Saint Pierre And Miquelon, Saint Vincent And The Grenadines, Samoa, San Marino, Sao Tome And Principe, Saudi Arabia, Senegal, Seychelles, Sierra Leone, Slovakia, Slovenia, Solomon Islands, Somalia, South Africa, Spain, Sri Lanka, Sudan, Surivalue, Swaziland, Sweden, Switzerland, Syria, Tahiti, Taiwan, Tajikistan, Tanzania, Thailand, The Netherlands, Togo, Tonga, Trinidad And Tobago, Tristan Da Cunha, Tunisia, Turkey, Turkmenistan, Turks And Caicos Islands, Tuvalu, Uganda, Ukraine, United Arab Emirates, United Kingdom, United States, Uruguay, Uzbekistan, Vanuatu, Venezuela, Vietnam, Virgin Islands (US), Wallis And Futuna, Yemen, Yugoslavia, Zambia, Zimbabwe

Parameter	Description
Option.OutputCountryFormat	Specifies the format to use for the country name returned in the **Country** output field. For example, if you select English, the country name "Deutschland" would be returned as "Germany". **E** Use English country names (default). **I** Use two-letter ISO abbreviation for the countries instead of country names. **U** Use Universal Postal Union abbreviation for the countries instead of country names.
Option.OutputScript	Specifies the alphabet or script in which the output should be returned. This option is bi-directional and generally takes place from Native to Latin and Latin to Native. **Input** Do not perform transliteration and provide output in the same script as the input (default). **Native** Output in the native script for the selected country wherever possible. **Latn** Use English values.
Option.MaximumResults	The maximum number of candidate addresses that GetCandidateAddressesLoqate should return. The default is 10. The maximum is 99.

Response

GetCandidateAddressesLoqate returns the following output.

Table 74: GetCandidateAddressesLoqate Output

Response Element	Description
AddressLine1	The first address line.
AddressLine2	The second address line.
AddressLine3	The third address line.
AddressLine4	The fourth address line.
City	The city name.
Country	The three-character ISO 3116-1 Alpha-3 code for the country. For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
FirmName	The firm name.
PostalCode	The postal code. In the U.S. this is the ZIP Code™.
PostalCode.AddOn	The last four digits of the ZIP + 4® Code. U.S. addresses only.
ProcessedBy	Indicates which address coder processed the address. **LOQATE** The Loqate coder processed the address.

Response Element	Description
StateProvince	The state or province abbreviation.
Status	Reports the success or failure of the match attempt. **null** Success **F** Failure
Status.Code	The reason for failure, if there is one. There is only one possible value: • RequestFailed
Status.Description	A description of the problem, if there is one. There is only one possible value: **Did not return multiples** The input address matched only one address in the database. GetCandidateAddressesLoqate only returns data if multiple possible matches were found.

GetCityStateProvince

GetCityStateProvince returns a city and state/province for a given input postal code.

Note: GetCityStateProvince works with U.S. and Canadian addresses only.

GetCityStateProvince is part of the Universal Addressing Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/GetCityStateProvince/results.json
```

XML endpoint:

```
http://server:port/rest/GetCityStateProvince/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GetCityStateProvince/results.json?
Data.PostalCode=12180
```

The JSON returned by this request would be:

```
{"output_port": [{
   "ProcessedBy": "USA",
   "PostalCode": "12180",
   "City": "TROY",
   "StateProvince": "NY",
   "Country": "USA",
   "City.Type": "P",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GetCityStateProvince/results.xml?
Data.PostalCode=12180
```

The XML returned by this request would be:

```
<ns2:xml.GetCityStateProvinceResponse
xmlns:ns2="http://www.pb.com/spectrum/services/GetCityStateProvince">
   <ns2:output_port>
      <ns2:Result>
         <ns2:ProcessedBy>USA</ns2:ProcessedBy>
         <ns2:PostalCode>12180</ns2:PostalCode>
         <ns2:City>TROY</ns2:City>
         <ns2:City.Type>P</ns2:City.Type>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:Country>USA</ns2:Country>
         <ns2:user_fields/>
      </ns2:Result>
   </ns2:output_port>
</ns2:xml.GetCityStateProvinceResponse>
```

Request

Parameters for Input Data

The following table shows the input fields.

Table 75: GetCityStateProvince Input

Parameter	Description
Data.PostalCode	A U.S. ZIP Code™ or Canadian postal code in one of the following formats: 99999 99999-9999 A9A9A9 A9A 9A9

Parameters for Options

Table 76: GetCityStateProvince Options

Parameter Name	Description
Option.PerformUSProcessing	Specifies whether or not to process U.S. addresses. If you enable U.S. address processing GetCityStateProvince will attempt to return the state for U.S. addresses. If you disable U.S. address processing, U.S. addresses will fail, meaning they are returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for U.S. address processing you must disable U.S. address processing in order for your jobs to complete successfully, regardless of whether or not they contain U.S. addresses.

Parameter Name	Description
	Note: You must have a valid license for U.S. address processing to successfully process U.S. addresses. If you enable U.S. address processing but are not licensed for this feature, or your license has expired, you will receive an error. **Y** Yes, process U.S. addresses (default). **N** No, do not process U.S. addresses.
Option.Database.US	Specifies the database to be used for U.S. address processing. Only databases that have been defined in the **US Database Resources** panel in the Management Console are available.
Option.PerformCanadianProcessing	Specifies whether or not to process Canadian addresses. If you enable Canadian address processing GetCityStateProvince will attempt to return the province for Canadian addresses. If you disable Canadian address processing, Canadian addresses will fail, meaning they are returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for Canadian address processing you must disable Canadian address processing in order for your jobs to complete successfully, regardless of whether or not they contain Canadian addresses. **Note:** You must have a valid license for Canadian address processing to successfully process Canadian addresses. If you enable Canadian address processing but are not licensed for this feature, or your license has expired, you will receive an error. **Y** Yes, process Canadian addresses (default). **N** No, do not process Canadian addresses.
Option.Database.Canada	Specifies the database to be used for Canadian address processing. Only databases that have been defined in the **Canadian Database Resources** panel in the Management Console are available.
Option.OutputVanityCity	Specifies whether or not to include non-mailing city names in the output. A non-mailing city name is an alternate name for the primary city name. For example, Hollywood is a non-mailing city name for Los Angeles. **Y** Yes, include non-mailing city names. **N** No, do not include non-mailing city names (default).
Option.MaximumResults	Specifies the maximum number of city-state/province pairs to return. The default value is 10.

Response

GetCityStateProvince returns the matching city and state/province for the input postal code as well as a code to indicate the success or failure of the match attempt. If more than one city/state or city/province matches the input postal code, multiple output records are returned.

Table 77: GetCityStateProvince Output

Response Element	Description
City	The matched city name.
City.Type	The USPS® standardized city name type (U.S. addresses only). **V** Vanity (non-mailing) city name. **P** Primary. The city name is the primary mailing city name. **S** Secondary. The city name is an alternate city name but is acceptable. A city can have multiple secondary city names.
PostalCode	The input postal code.
ProcessedBy	Indicates which address coder processed the address. One of the following: **USA** The U.S. address coder processed the address. **CAN** The Canadian address coder processed the address.
StateProvince	The state or province abbreviation.
Status	Reports the success or failure of the match attempt. **null** Success **F** Failure
Status.Code	The reason for failure, if there is one. The only valid value is: • DisabledCoder • UnrecognizedPostalCode
Status.Description	The description of the failure. The valid values are: **Postal code not found** This value will appear if Status.Code=UnrecognizedPostalCode. **PerformUSProcessing disabled** This value will appear if Status.Code=DisabledCoder. **PerformCanadianProcessing disabled** This value will appear if Status.Code=DisabledCoder.

GetCityStateProvinceLoqate

GetCityStateProvinceLoqate returns a city and state/province for a given input postal code.

This stage is part of the Universal Addressing Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/GetCityStateProvinceLoqate/results.json
```

XML endpoint:

```
http://server:port/rest/GetCityStateProvinceLoqate/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GetCityStateProvinceLoqate/results.json?
Data.Country=USA&Data.PostalCode=60510
```

The JSON returned by this request would be:

```
{"output_port": [{
   "ProcessedBy": "LOQATE",
   "PostalCode": "60510",
   "City": "Batavia",
   "StateProvince": "IL",
   "Country": "United States",
   "Status": "",
   "Status.Code": "",
   "Status.Description": "",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GetCityStateProvinceLoqate/results.xml?Data.Country=USA&
Data.PostalCode=60510
```

The XML returned by this request would be:

```
<ns2:xml.GetCityStateProvinceLoqateResponse
xmlns:ns2="http://www.pb.com/spectrum/services/GetCityStateProvinceLoqate">

   <ns2:output_port>
      <ns2:Result>
         <ns2:ProcessedBy>LOQATE</ns2:ProcessedBy>
         <ns2:PostalCode>60510</ns2:PostalCode>
         <ns2:City>Batavia</ns2:City>
         <ns2:StateProvince>IL</ns2:StateProvince>
         <ns2:Country>United States</ns2:Country>
         <ns2:Status/>
         <ns2:Status.Code/>
         <ns2:Status.Description/>
         <ns2:user_fields/>
      </ns2:Result>
   </ns2:output_port>
</ns2:xml.GetCityStateProvinceLoqateResponse>
```

Request

Parameters for Input Data

The following table shows the input fields.

Table 78: GetCityStateProvinceLoqate Input

Parameter	Description
Data.Country	The country code or name, in any of the following formats: • 2-digit ISO country code • 3-digit UPU Country code • English country name For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Data.PostalCode	The postal code for the address.

Options

Table 79: GetCityStateProvinceLoqate Options

	Description / Valid Values
	Specifies the database to be used for address processing. Only databases that have been defined in the **Database Resources** panel in the Management Console are available.
	The maximum number of addresses that GetCityStateProvinceLoqate should return. The default is 10.
	Specifies the alphabet or script in which the output should be returned. This option is bi-directional and generally takes place from Native to Latin and Latin to Native. Do not perform transliteration and provide output in the same script as the input (default). Output in the native script for the selected country wherever possible. Use English values.
	Specifies how you want Spectrum Technology Platform to respond when a data license error occurs. **Fail the job** Fail the entire job if a data license error occurs. **Fail the record** Fail the record(s) for which the data license error occurs and continue processing.

Response

GetCityStateProvinceLoqate returns the matching city and state/province for the input postal code as well as a code to indicate the success or failure of the match attempt. If more than one city/state or city/province matches the input postal code, multiple output records are returned.

Table 80: GetCityStateProvinceLoqate Output

Response Element	Description
City	The matched city name.
Country	The country in the format determined by what you selected in: • ISO Code • UPU Code • English
PostalCode	The input postal code.
ProcessedBy	Indicates which address coder processed the address. **LOQATE** The Loqate coder processed the address.
StateProvince	The state or province abbreviation.
Status	Reports the success or failure of the match attempt. **null** Success **F** Failure
Status.Code	The reason for failure, if there is one. The only valid value is: • UnrecognizedPostalCode
Status.Description	The description of the failure. The only valid value is: **Postal code not found** This value will appear if Status.Code=UnrecognizedPostalCode.

GetPostalCodes

GetPostalCodes allows you to look up the postal codes for a particular city. The service takes a city, state, and country as input and returns the postal codes for that city. The input must be exactly correct in order to return postal codes.

Note: GetPostalCodes only works with U.S. addresses.

GetPostalCodes is part of the Universal Addressing Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/GetPostalCodes/results.json
```

XML endpoint:

```
http://server:port/rest/GetPostalCodes/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/GetPostalCodes/results.json?
Data.City=Holland&Data.StateProvince=MI
```

The JSON returned by this request would be:

```
{"output_port": [
      {
      "ProcessedBy": "USA",
      "PostalCode": "49422",
      "Status": "",
      "City.Type": " ",
      "Status.Code": "",
      "Status.Description": "",
      "user_fields": []
   },
      {
      "ProcessedBy": "USA",
      "PostalCode": "49423",
      "Status": "",
      "City.Type": " ",
      "Status.Code": "",
      "Status.Description": "",
      "user_fields": []
   },
      {
      "ProcessedBy": "USA",
      "PostalCode": "49424",
      "Status": "",
      "City.Type": " ",
      "Status.Code": "",
      "Status.Description": "",
      "user_fields": []
   }
]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/GetPostalCodes/results.xml?Data.City=Holland&
Data.StateProvince=MI
```

The XML returned by this request would be:

```
<ns2:xml.GetPostalCodesResponse
xmlns:ns2="http://www.pb.com/spectrum/services/GetPostalCodes">
   <ns2:output_port>
      <ns2:Result>
         <ns2:ProcessedBy>USA</ns2:ProcessedBy>
         <ns2:PostalCode>49422</ns2:PostalCode>
         <ns2:City.Type></ns2:City.Type>
         <ns2:Status/>
         <ns2:Status.Code/>
         <ns2:Status.Description/>
         <ns2:user_fields/>
      </ns2:Result>
      <ns2:Result>
         <ns2:ProcessedBy>USA</ns2:ProcessedBy>
         <ns2:PostalCode>49423</ns2:PostalCode>
         <ns2:City.Type></ns2:City.Type>
         <ns2:Status/>
         <ns2:Status.Code/>
         <ns2:Status.Description/>
         <ns2:user_fields/>
      </ns2:Result>
      <ns2:Result>
         <ns2:ProcessedBy>USA</ns2:ProcessedBy>
         <ns2:PostalCode>49424</ns2:PostalCode>
         <ns2:City.Type></ns2:City.Type>
         <ns2:Status/>
         <ns2:Status.Code/>
         <ns2:Status.Description/>
         <ns2:user_fields/>
      </ns2:Result>
   </ns2:output_port>
</ns2:xml.GetPostalCodesResponse>
```

Request

Parameters for Input Data

GetPostalCodes takes a city, state/province, and country as input.

Table 81: GetPostalCodes Input

Parameter	Description
Data.City	The city whose postal codes you want to look up. You may put the city and state in the City field. If you do this, you must leave the StateProvince field blank. The total length of the City and StateProvince fields cannot exceed 100 characters.
Data.StateProvince	The state or province of the city whose postal codes you want to look up. You may also put the state in the City field instead of the StateProvince field. The total length of the City and StateProvince fields cannot exceed 100 characters.
Data.Country	The country code or name of the city whose postal codes you want to look up. The only valid value is US.

Parameters for Options

Table 82: GetPostalCodes Options

Parameter	Description
Option.Database.US	Specifies the database to be used for postal code look-ups. Only databases that have been defined in the US Database Resources panel in the Management Console are available.
Option.IncludeVanityCity	Specifies whether or not to include postal codes for the city's non-mailing city names. A non-mailing city name is an alternate name for the primary city name. For example, Hollywood is a non-mailing city name for Los Angeles. **Y** Yes, include postal codes for non-mailing city names.

Parameter	Description
	N No, do not include postal codes for non-mailing city names (default).
Option.OutputCityType	Specifies whether or not to return the city type in the output. If enabled, the city type is returned in the City.Type field. **Y** Yes, include the city type in the output. **N** No, do not include the city type in the output (default).

Response

GetPostalCodes returns the postal codes for a specified city. Each postal code is returned in a separate record along with the data listed in the following table.

Table 83: GetPostalCodes Output

Response Element	Description
City.Type	The USPS® city type (U.S. addresses only). The city type is determined by looking at the ZIP Code and the city name. For example, the city Lanham MD has the postal codes 20703, 20706, and 20784. Lanham is the primary city in 20703 and 20706 but is a vanity city in 20784. This field column is only populated if . The possible values are: **V** Vanity (non-mailing) city name. **P** Primary. The city name is the primary mailing city name. **S** Secondary. The city name is an alternate city name but is acceptable. A city can have multiple secondary city names.
PostalCode	A postal code in the specified city.
ProcessedBy	Because this service only works for U.S. addresses, ProcessedBy will always contain one value: USA.
Status	Reports the success or failure of the match attempt. **null** Success **F** Failure
Status.Code	Reason for failure, if there is one. One of the following: • CountryNotSupported • UnableToLookup
Status.Description	Description of failure. • Input country is not supported • Input city was blank • Input city & state / province was blank, or no match found • City-state mismatch (different spelling found, or city-state was a vanity name and vanity matching was not allowed, or city-state did not match ZIP Code)

ValidateAddress

ValidateAddress standardizes and validates addresses using postal authority address data. ValidateAddress can correct information and format the address using the format preferred by the applicable postal authority. It also adds missing postal information, such as postal codes, city names, state/province names, and more.

ValidateAddress also returns result indicators about validation attempts, such as whether or not ValidateAddress validated the address, the level of confidence in the returned address, the reason for failure if the address could not be validated, and more.

During address matching and standardization, ValidateAddress separates address lines into components and compares them to the contents of the Universal Addressing Module databases. If a match is found, the input address is *standardized* to the database information. If no database match is found, ValidateAddress optionally *formats* the input addresses. The formatting process attempts to structure the address lines according to the conventions of the appropriate postal authority.

ValidateAddress is part of the Universal Addressing Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/ValidateAddress/results.json
```

XML endpoint:

```
http://server:port/rest/ValidateAddress/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/ValidateAddress/results.json?Data.AddressLine1=
1825+Kramer+Ln&Data.PostalCode=78758
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Confidence": "100",
   "RecordType": "Normal",
   "CountryLevel": "A",
   "ProcessedBy": "USA",
   "MatchScore": "0",
   "AddressLine1": "1825 Kramer Ln",
   "City": "Austin",
   "StateProvince": "TX",
   "PostalCode": "78758-4260",
   "Country": "United States Of America",
   "PostalCode.Base": "78758",
   "PostalCode.AddOn": "4260",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/ValidateAddress/results.xml?Data.AddressLine1=
1825+Kramer+Ln&Data.PostalCode=78758
```

The XML returned by this request would be:

```
<ns2:xml.ValidateAddressResponse
xmlns:ns2="http://www.pb.com/spectrum/services/ValidateAddress">
   <ns2:output_port>
      <ns2:Address>
         <ns2:Confidence>93</ns2:Confidence>
         <ns2:RecordType>Normal</ns2:RecordType>
         <ns2:CountryLevel>A</ns2:CountryLevel>
         <ns2:ProcessedBy>USA</ns2:ProcessedBy>
         <ns2:MatchScore>0</ns2:MatchScore>
         <ns2:AddressLine1>1825 Kramer Ln</ns2:AddressLine1>
         <ns2:City>Austin</ns2:City>
         <ns2:StateProvince>TX</ns2:StateProvince>
         <ns2:PostalCode>78758-4260</ns2:PostalCode>
         <ns2:PostalCode.Base>78758</ns2:PostalCode.Base>
         <ns2:PostalCode.AddOn>4260</ns2:PostalCode.AddOn>
         <ns2:Country>United States Of America</ns2:Country>
         <ns2:user_fields/>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.ValidateAddressResponse>
```

Request

Parameters for Input Data

ValidateAddress takes an address as input. All addresses use this format regardless of the address's country. See **Address Line Processing for U.S. Addresses** on page 183 for important information about how address line data is processed for U.S. addresses.

Table 84: Input Format

Parameter	Format	Description
Data.AddressLine1	String [50]	The first address line.
Data.AddressLine2	String [50]	The second address line.
Data.AddressLine3	String [50]	The third address line. Does not apply to Canadian addresses.
Data.AddressLine5	String [50]	The fourth address line. Does not apply to Canadian addresses.
Data.AddressLine5	String [50]	The fifth address line. Applies only to U.K. addresses. May contain street name, unit number, building number, and so on.
Data.City	String [50]	The city name. For U.S. addresses only, you may put the city, state, and ZIP Code™ in the City field. If you do this, you must leave the StateProvince and PostalCode fields blank.
Data.StateProvince	String [50]	The state or province. For U.S. addresses only, you may put the state in the City field instead of the StateProvince field.

Parameter	Format	Description
Data.PostalCode	String [10]	The postal code for the address in one of the following formats: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999 For U.S. addresses only, you may put the ZIP Code™ in the City field. For U.S. addresses only, if the city/state/ZIP Code™ is in the PostalCode field, ValidateAddress may parse the data and successfully process the address. For best results, put this data in the appropriate fields (City, StateProvince, and PostalCode).
Data.Country	String [50]	The country code or name, in any of the following formats: • Two-character ISO 3116-1 Alpha-2 country code • Three-character ISO 3116-1 Alpha-3 country code • English country name • French country name • German country name • Spanish country name For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Data.FirmName	String [50]	The company or firm name.
Data.USUrbanName	String [50]	The U.S. address urbanization name. This is used primarily for Puerto Rico addresses.
Data.CustomerID	String [9]	If this mailpiece uses a generic barcode, specify your USPS®-assigned customer ID in this field. The ValidateAddress generic barcode is used for mailpieces that use the OneCode ACS® service.
Data.CanLanguage	String	For Canadian addresses only, indicates whether the address is in English or French, if the option Option.CanFrenchFormat=T is used. If this field is blank, the address is formatted in English. If the field contains any non-blank value, the address is formatted in French. Note that addresses in Quebec are always formatted in French regardless of the value in this field.

Address Line Processing for U.S. Addresses

The input fields AddressLine1 through AddressLine4 are handled differently for U.S. addresses depending on whether the firm name extraction or urbanization code extraction options are enabled. If either of these options is enabled, ValidateAddress will look at the data in all four fields to validate the address and extract the requested data (firm name and/or urbanization code). If neither of these options is enabled, ValidateAddress uses only the first two non-blank address line fields in its validation attempt. The data in the other address line fields is returned in the output field AdditionalInputData. For example,

AddressLine1: A1 Calle A
AddressLine2:
AddressLine3: URB Alamar
AddressLine4: Pitney Bowes Software

In this address, if either firm name extraction or urbanization code extraction were enabled, ValidateAddress would examine all four address lines. If neither firm name extraction nor urbanization code extraction were enabled, ValidateAddress would examine AddressLine1 and AddressLine3 (the first two non-blank address lines) and attempt to validate the address using that data; the data in AddressLine4 would be returned in the output field AdditionalInputData.

Parameters for Options

Output Data Options

The following table lists the options that control the type of information returned by ValidateAddress. Some of these options can be overridden for Canadian addresses. For more information, see **Canadian Address Options** on page 202.

Table 85: Output Data Options

Parameter	Description
Option.OutputRecordType	Type of output record. For more than one, provide a list. **A** Returns 1 to 4 lines of address data plus city, state, postal code, firm name, and urbanization name information. Each address line represents an actual line of the address as it would appear on an envelope. For more information, see **Response** on page 211. If ValidateAddress could validate the address, the address lines contain the standardized address. When addresses are standardized, punctuation is removed, directionals are abbreviated, street suffixes are abbreviated, and address elements are corrected. If ValidateAddress could not validate the address, the address lines contain the address as it appeared in the input ("pass through" data). Non-validated addresses are always included as pass through data in the address line fields even if you do not specify **OutputRecordType=A**. **E** Parsed address elements. Each part of the address, such as house number, street name, street suffix, directionals, and so on is returned in a separate field. For more information, see **Parsed Address Elements Output** on page 212. Note that if you specify "E" and specify **OutputFormattedOnFail=Y**, the parsed address elements will contain the input address for addresses that could not be validated. **I** Parsed input. This option returns the input address in parsed form regardless of whether or not ValidateAddress is able to validate the address. Each part of the input address, such as house number, street name, street suffix, directionals, and so on is returned in a separate field. Parsed input (value "I") differs from the combination of **OutputRecordType=E** and **OutputFormattedOnFail=Y** in that "I" returns all input address in parsed form, not just input that could

Parameter	Description
	not be validated. For more information, see **Parsed Input** on page 213. **P** Postal data. Output addresses contain additional data for each validated address. For more information, see **Postal Data Output** on page 215. **Blank** Do not return any address data or postal data.
Option.OutputFieldLevelReturnCodes	Specifies whether to include field-level result indicators. Field-level result indicators describe how ValidateAddress handled each address element. Field-level result indicators are returned in the qualifier "Result". For example, the field-level result indicator for HouseNumber is contained in **HouseNumber.Result**. For a complete listing of result indicator output fields, see **Field-Level Result Indicators** on page 220. **N** No, do not output field-level return codes (default). **Y** Yes, output field-level return codes.
Option.OutputFormattedOnFail	Specifies whether to return a formatted address when an address cannot be validated. The address is formatted using the preferred address format for the address's country. If this option is not selected, the output address fields are blank when ValidateAddress cannot validate the address. **Note:** This option applies only to U.S. and Canadian addresses. Formatted data will not be returned for any other address. **N** No, do not format failed addresses (default). **Y** Yes, format failed addresses. Formatted addresses are returned using the format specified by the OutputRecordType option. Note that if you specify OutputRecordType=E, the parsed address elements will contain the parsed, validated address for addresses that could be validated. If the address could not be validated the parsed address elements will contain the input address in parsed form. If you always want the output to contain the input address in parsed form, regardless of whether or not ValidateAddress could validate the address, specify OutputRecordType=I. Formatted addresses are returned using the format specified by the Option.OutputRecordType option. Note that if you specify Option.OutputRecordType=E, the parsed address elements will contain the parsed, validated address for addresses that could be validated. If the address could not be validated the parsed address elements will contain the input address in parsed form. If you always want the output to contain the input address in parsed form, regardless of whether or not ValidateAddress could validate the address, specify Option.OutputRecordType=I.

Parameter	Description
	Formatted addresses are returned using the format specified by the **Include a standard address**, **Include address line elements**, and **Include postal information** check boxes. Note that if you select **Include address line elements**, the parsed address elements will contain the parsed, validated address for addresses that could be validated. If the address could not be validated the parsed address elements will contain the input address in parsed form. If you always want the output to contain the input address in parsed form, regardless of whether or not ValidateAddress could validate the address, select **Include standardized input address elements**. If you specify Y, you must specify "A" and/or "E" for OutputRecordType. If you specify Y, you must specify "A" and/or "E" for Option.OutputRecordType. If you check this option, you must select **Include a standard address** and/or **Include address line elements**.
Option.OutputStreetNameAlias	For U.S. addresses only, specifies whether or not to use a street's alias in the output. A street alias is an alternate name for a street and typically applies only to a specific range of addresses on the street. If you do not allow street aliases in the output then the street's "base" name will appear in the output regardless of whether or not there is an alias for the street. The base name is the name that applies to the entire street. **N** No, do not return street name aliases in the output. **Y** Yes, return street name aliases in the output if there is an alias for the street (default).
Option.OutputStreetNameAlias	For U.S. addresses only, specifies how to handle street name aliases used in the input. A street alias is an alternate name for a street and typically applies only to a specific range of addresses on the street. If you enable this option, street name aliases used in the input will appear in the output. If you do not enable this option, street name aliases in the input will be converted to the base street name in the output, with the following exceptions: • If a preferred alias is used in input the preferred alias will always be used in output. • Changed aliases used in input are always converted to the base street name in output. This is one of three options that control how ValidateAddress handles street name aliases. The other two are Option.OutputPreferredAlias and Option.OutputAbbreviatedAlias. **Note:** If Option.OutputAbbreviatedAlias is enabled, the abbreviated alias will always appear in the output

Parameter	Description
	even if you have Option.OutputStreetNameAlias disabled. **N** No, do not return street name aliases in the output. **Y** Yes, return street name aliases in the output if the input street name is an alias (default).
Option.OutputAddressBlocks	Specifies whether to return a formatted version of the address as it would be printed on a physical mailpiece. Each line of the address is returned in a separate address block field. There can be up to nine address block output fields: **AddressBlock1** through **AddressBlock9**. For example, this input address: AddressLine1: 4200 Parliament Place AddressLine2: Suite 600 City: Lanham StateProvince: MD PostalCode: 20706 Results in this address block output: AddressBlock1: 4200 PARLIAMENT PL STE 600 AddressBlock2: LANHAM MD 20706-1882 AddressBlock3: UNITED STATES OF AMERICA ValidateAddress formats the address into address blocks using postal authority standards. The country name is returned using the Universal Postal Union country name. Note that the option Option.OutputCountryFormat does not affect the country name in the address block, it only affects the name returned in the **Country** output field. For addresses outside the U.S. and Canada, if ValidateAddress is unable to validate the address, no address blocks are returned. For addresses in the U.S. and Canada, address blocks are returned even if validation fails. **N** No, do not return address blocks. Default. **Y** Yes, return address blocks.

Obtaining Congressional Districts

ValidateAddress can determine the U.S. congressional district for an address.

To obtain congressional districts, `Option.OutputRecordType` must contain P. For more information on `Option.OutputRecordType`, see **Output Data Options** on page 184.

Table 86: Congressional District Output

Response Element	Description
USCongressionalDistrict	Congressional district number. If the address is a non-state address (for example Puerto Rico or Washington D.C.) this field is blank.

Obtaining County Names

ValidateAddress can determine the county where a particular address is located and return the county name.

Note: County names are available for U.S. addresses only.

To obtain county names, `Option.OutputRecordType` must contain P. For more information on `Option.OutputRecordType`, see **Output Data Options** on page 184.

Table 87: County Name Output

Response Element	Description
USCountyName	County name

Obtaining FIPS County Numbers

Federal Information Processing Standards (FIPS) county numbers are numbers that identify each county in a state. Note that these numbers are only unique at the state level, not the national level. For more information, see **http://www.census.gov**.

Note: FIPS county numbers are available for U.S. addresses only.

To obtain FIPS county numbers, `Option.OutputRecordType` must contain P. For more information on `Option.OutputRecordType`, see **Output Data Options** on page 184.

Table 88: FIPS County Number Output

Response Element	Description
USFIPSCountyNumber	FIPS (Federal Information Processing Standards) county number

Obtaining Carrier Route Codes

Carrier route codes are unique identifiers assigned to each mail carrier who delivers mail, allowing unique identification of each U.S. delivery route. ValidateAddress can return the code that represents an addressee's carrier route.

Note: Carrier route codes are available for U.S. addresses only.

To obtain carrier route codes, `Option.OutputRecordType` must contain P. For more information on `Option.OutputRecordType`, see **Output Data Options** on page 184.

Table 89: Carrier Route Code Output

Response Element	Description
USCarrierRouteCode	Carrier route code

Creating Delivery Point Barcodes

A Delivery Point Barcode (DPBC) is a POSTNET™ barcode representation of the address. It consists of 62 bars with beginning and ending frame bars and five bars each for the ZIP + 4® Code, a value calculated based on the street address number, and a correction digit. The DPBC allows automated sortation of letter mail to the carrier level in walk sequence. ValidateAddress generates the data you need to assemble a DPBC.

Note: Delivery Point Barcodes are available for U.S. addresses only. For more information on Delivery Point Barcodes, see **http://www.usps.com**.

To generate the data needed to assemble a DPBC, `Option.OutputRecordType` must contain P. For more information on `Option.OutputRecordType`, see **Output Data Options** on page 184.

Table 90: Delivery Point Barcode Output

Response Element	Description
PostalBarCode	The delivery point portion of the delivery point barcode.
USBCCheckDigit	Check-digit portion of the 11-digit delivery point barcode.

To assemble a DPBC you concatenate the values found in the ValidateAddress output as follows:

PostalCode.Base + PostalCode.Addon + PostalBarcode + USBCCheckDigit

For example, if you have the following:

- **PostalCode.Base** = 49423
- **PostalCode.Addon** = 4506
- **PostalBarcode** = 29
- **USBCCheckDigit** = 2

The assembled barcode would be:

494234506292

Default Options

The following table lists the options that control the format and processing of addresses. These are called "default options" because by default the apply to all addresses. Some of these options can be overridden for Canadian addresses. For more information, see **Canadian Address Options** on page 202.

Table 91: Default Options

Parameter	Description
Option.OutputCasing	Specifies the casing of the output data. One of the following: **M** The output in mixed case (default). For example: 123 Main St Mytown FL 12345 **U** The output in upper case. For example: 123 MAIN ST MYTOWN FL 12345
Option.OutputPostalCodeSeparator	Specifies whether to use separators (spaces or hyphens) in ZIP™ Codes or Canadian postal codes. For example, a ZIP + 4® Code with the separator would be 20706-1844 and without the separator it would be 207061844. A Canadian postal code with the separator would be P5E"1S7 and without the separator it would be P5E1S7. **Y** Yes, use separator (default).

Parameter	Description
	N No, do not use separator. **Note:** Spaces are used in Canadian postal codes and hyphens in U.S. ZIP + 4® Codes.
Option.OutputMultinationalCharacters	Specifies whether or not to return multinational characters, including diacritical marks such as umlauts or accents. (Not supported for U.S. addresses). **N** No, do not use multinational characters in the output (default). Only standard ASCII characters is returned. **Y** Yes, use multinational characters in the output.
Option.KeepMultimatch	Indicates whether or not to return multiple address for those input addresses that have more than one possible match. **Y** Yes, return multiple matches (default). **N** No, do not return multiple matches. For more information, see **Returning Multiple Matches** on page 193.
Option.StandardAddressFormat	Specifies where to place secondary address information for U.S. addresses. Secondary address information refers to apartment numbers, suite numbers, and similar designators. For example, in this address the secondary address information is "Apt 10E" and the primary address information is "424 Washington Blvd". Apt 10E 424 Washington Blvd Springfield MI 49423 **C** Place both primary and secondary address information in AddressLine1 (default). **S** Place the primary address information in AddressLine1 and the secondary address information in AddressLine2. **D** Place both primary and secondary address information in AddressLine1 and place dropped information from dual addresses in AddressLine2. A dual address is an address that contains both street information and PO Box/Rural Route/Highway Contract information. For more information, see **About Dual Address Logic** on page 193.
Option.OutputShortCityName	Specifies how to format city names that have short city name or non-mailing city name alternatives. Applies to U.S. and Canadian addresses. **Y** Returns the USPS®-approved abbreviation for the city, if there is one. The USPS® provides abbreviations for city names that are 14 characters long or longer. City abbreviations are 13 characters or less and can be used when there is limited space on the mailing label. If there

Parameter	Description
	is no short city name for the city, then the full city name is returned. **N** Returns the long city name (default). **S** Returns the abbreviated city name only if an abbreviated city name is used in the input address. If the input address does not use a short city name, either the long or short city name could be returned, depending on USPS® regulations for the particular city. Select this option if you are performing a CASS™ test. **V** Output the non-mailing city name (the vanity name) if the input city name is a non-mailing city name. For example, "Hollywood" is a non-mailing city name for "Los Angeles". If you do not select this option and the input city name is a non-mailing city name the long version of the mailing city is returned.
Option.OutputCountryFormat	Specifies the format to use for the country name returned in the **Country** output field. For example, if you select English, the country name "Deutschland" would be returned as "Germany". **E** Use English country names (default). **S** Use Spanish country names. **F** Use French country names. **G** Use German country names. **I** Use two-letter ISO abbreviation for the countries instead of country names. **U** Use Universal Postal Union abbreviation for the countries instead of country names.
Option.HomeCountry	Specifies the default country. You should specify the country where most of your addresses reside. For example, if most of the addresses you process are in Canada, specify Canada. ValidateAddress uses the country you specify to attempt validation when it cannot determine the country from the StateProvince, PostalCode, and Country address fields. The valid country names are: Afghanistan, Albania, Algeria, American Somoa, Andorra, Angola, Anguilla, Antigua And Barbuda, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, Bosnia And Herzegovina, Botswana, Brazil, British Virgin Islands, Brunei Darussalam, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Cayman Islands, Central African Republic, Chad, Chile, China, Colombia, Comoros Islands, Congo, Cook Islands, Costa Rica, Cote D'Ivoire, Croatia, Cuba, Cyprus, Czech Republic, Democratic Republic Of Congo, Denmark, Djibouti, Dominica, Dominican Republic, East Timor, Ecuador, Egypt, El Salvador, Equitorial Guinea,

Parameter	Description
	Eritrea, Estonia, Ethiopia, Falkland Islands, Faroe Islands, Federated States Of Micronesia, Fiji, Finland, France, French Guiana, Gabon, Gambia, Germany, Ghana, Gibraltar, Greece, Greenland, Grenada, Guadeloupe, Guam, Guatemala, Guinea, Guinea Bissau, Guyana, Haiti, Holy See, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, Israel, Italy, Jamaica, Japan, Jordan, Kazakhstan, Kenya, Kiribati, Korea, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Marshall Islands, Martinique, Mauritania, Mauritius, Mayotte, Mexico, Moldova, Monaco, Mongolia, Monserrat, Morocco, Mozambique, Myanmar, Namibia, Nauru, Nepal, Netherlands Antilles, New Caledonia, New Zealand, Nicaragua, Niger, Nigeria, Niue, Norway, Oman, Pakistan, Palau, Panama, Papua New Guinea, Paraguay, Peru, Philippines, Pitcairn Islands, Poland, Portugal, Puerto Rico, Qatar, Republic Of Georgia, Republic Of Korea, Republic Of Singapore, Reunion, Romania, Russia, Rwanda, Saint Helena, Saint Kitts And Nevis, Saint Lucia, Saint Pierre And Miquelon, Saint Vincent And The Grenadines, Samoa, San Marino, Sao Tome And Principe, Saudi Arabia, Senegal, Seychelles, Sierra Leone, Slovakia, Slovenia, Solomon Islands, Somalia, South Africa, Spain, Sri Lanka, Sudan, Surivalue, Swaziland, Sweden, Switzerland, Syria, Tahiti, Taiwan, Tajikistan, Tanzania, Thailand, The Netherlands, Togo, Tonga, Trinidad And Tobago, Tristan Da Cunha, Tunisia, Turkey, Turkmenistan, Turks And Caicos Islands, Tuvalu, Uganda, Ukraine, United Arab Emirates, United Kingdom, United States, Uruguay, Uzbekistan, Vanuatu, Venezuela, Vietnam, Virgin Islands (US), Wallis And Futuna, Yemen, Yugoslavia, Zambia, Zimbabwe
Option.DualAddressLogic	Indicates how ValidateAddress should return a match if multiple non-blank address lines are present or multiple address types are on the same address line (U.S. addresses only). **N** (Default) USPS® CASS™ regulations determine the address returned based on the following order of priority: **1.** PO Box **2.** Firm **3.** Highrise **4.** Street **5.** Rural Route **6.** General Delivery **S** Return a street match, regardless of the address line. **P** Return a PO Box match, regardless of the address line. For more information, see **About Dual Address Logic** on page 193.

About Dual Address Logic

For U.S. addresses only, the Option.DualAddressLogic option controls whether ValidateAddress should return a street match or a PO Box/Rural Route/Highway Contract match when the address contains both street and PO Box/Rural Route/Highway Contract information in the same address line.

Note: The Option.DualAddressLogic option has no effect if the street information is in a different address line input field than the PO Box/Rural Route/Highway Contract information.

For example, given the following input address:

AddressLine1: 401 N Main St Apt 1 POB 1
City: Kemp
StateProvince: TX
PostalCode: 75143

ValidateAddress would return one of the following:

- If Option.DualAddressLogic is set to either N or P, ValidateAddress returns the following:

  AddressLine1: PO Box 1
  City: Kemp
  StateProvince: TX
  PostalCode: 75143-0001

- If Option.DualAddressLogic is set to S, ValidateAddress returns the following:

  AddressLine1: 401 N Main St Apt 1
  City: Kemp
  StateProvince: TX
  PostalCode: 75143-4806

The address data that is not used to standardize the address can be returned in one of two places:

- **AddressLine2**—The address information not used to standardize the address is returned in the **AddressLine2** field if you specify Option.StandardAddressFormat=D. For more information on this option, see **Default Options** on page 189. For example, if you choose to return a street match for dual addresses,

  AddressLine1: 401 N Main St Apt 1
  AddressLine2: PO Box 1
  City: Kemp
  StateProvince: TX
  PostalCode: 75143-0001

- **AdditionalInputData**—If you do not specify Option.StandardAddressFormat=D then the address information not used to standardize the address is returned in the **AdditionalInputData** field. For more information on this option, see **Default Options** on page 189. For example, if you choose to return a street match for dual addresses,

  AddressLine1: 401 N Main St Apt 1
  City: Kemp
  StateProvince: TX
  PostalCode: 75143-0001
  AdditionalInputData: PO Box 1

Address information that is dropped can be retrieved by setting the Option.StandardAddressFormat option to D. For more information, see .

Returning Multiple Matches

If ValidateAddress finds multiple address in the postal database that are possible matches for the input address, you can have ValidateAddress return the possible matches. For example, the following address matches multiple addresses in the U.S. postal database:

PO BOX 1
New York, NY

Options

To return multiple matches, use the options described in the following table.

Table 92: Multiple Match Option

Parameter	Description
Option.KeepMultimatch	Indicates whether or not to return multiple address for those input addresses that have more than one possible match. **Y** Yes, return multiple matches (default). **N** No, do not return multiple matches.
Option.MaximumResults	A number between 1 and 10 that indicates the maximum number of addresses to return. The default value is 1. **Note:** The difference between Option.Keepmultimatch=N and Option.KeepMultimatch=Y/Option.MaximumResults=1 is that a multiple match will return a failure if Option.KeepMultimatch=N, whereas a multiple match will return one record if Option.KeepMultimatch=Y and Option.MaximumResults=1.
Option.OutputFieldLevelReturnCodes	To identify which output addresses are candidate addresses, you must specify a value of **Y** for Option.OutputFieldLevelReturnCodes. When you do this, records that are candidate addresses will have one or more "M" values in the field-level result indicators.

Output

When you choose to return multiple matches, the addresses are returned in the address format you specify. For information on specifying address format, see **Output Data Options** on page 184. To identify which records are the candidate addresses, look for multiple "M" values in the field-level result indicators. For more information, see **Field-Level Result Indicators** on page 220.

U.S. Address Options

Parameter	Description
Option.PerformUSProcessing	Specifies whether to process U.S. addresses. If you enable U.S. address processing ValidateAddress will attempt to validate U.S. addresses. If you disable U.S. address processing, U.S. addresses will fail, meaning they are returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for U.S. address processing you must disable U.S. address processing in order for your jobs to complete successfully, regardless of whether or not they contain U.S. addresses. **Note:** You must have a valid license for U.S. address processing to successfully process U.S. addresses. If you enable U.S. address processing but are not licensed for this feature, or your license has expired, you will receive an error. **N** No, do not process U.S. addresses. **Y** Yes, process U.S. addresses. Default.

Parameter	Description
Option.Database.US	Specifies which database to use for validating U.S. addresses. Only databases that have been defined in the US Database Resources panel in the Management Console are available.
Option.PerformLOT	Enhanced Line of Travel (eLOT) processing assigns a Line of Travel sequence code to your addresses. Note that ValidateAddress does not sort into eLOT sequence but it provides data (the Line of Travel sequence code) that allows you to sort addresses into eLOT sequence. To perform eLOT processing you must have the eLOT database installed. **N** No, do not perform Line of Travel Processing. Default. **Y** Yes, perform Line of Travel processing. For a listing of the output fields returned by this option, see **Enhanced Line of Travel Output** on page 230.
Option.PerformRDI	Residential Delivery Indicator (RDI™) processing checks if an address is a residential address (not a business address). To perform RDI™ processing, you must have the RDI™ database installed. If you enable both DPV® and RDI™ processing, RDI™ information is only returned if the address is a valid delivery point. If DPV® does not validate the address no RDI™ data is returned. **N** No, do not perform Residential Delivery Indicator processing. Default. **Y** Yes, perform Residential Delivery Indicator processing.
Option.PerformESM	Enhanced Street Matching (ESM) applies additional matching logic to correct misspelled or complex street names and obtain a match. ESM enables ValidateAddress to validate more addresses but it reduces performance. You cannot perform ESM when ASM is enabled. **N** No, do not perform enhanced street matching. Default. **Y** Yes, perform enhanced street matching.
Option.PerformASM	All Street Matching (ASM) applies ESM processing as well as additional matching logic to correct errors in street names and obtain a match. It is effective at matching streets when the first letter of the street is incorrect. ASM provides the best address validation but reduces performance. **N** No, do not perform all street matching. **Y** Yes, perform all street matching. Default.
Option.PerformDPV	Delivery Point Validation (DPV®) validates that a specific address exists, as opposed to validating that a specific address is within a range of valid addresses. CMRA processing checks if an address is for a mailbox rented from a private company, referred to as a Commercial Mail Receiving Agent (CMRA). To perform DPV and CMRA processing, you must have the DPV database installed. The DPV database contains both DPV and CMRA data.

Parameter	Description
	N No, do not perform Delivery Point Validation or CMRA processing. Default. **Y** Yes, perform Delivery Point Validation and CMRA processing. For a listing of the output fields returned by this option, see **DPV and CMRA Output** on page 232.
Option.PerformLACSLink	The USPS® Locatable Address Conversion System (LACS) allows you to correct addresses that have changed as a result of a rural route address converting to street-style address, a PO Box renumbering, or a street-style address changing. When enabled, LACS^{Link} processing is attempted for addresses that could not be validated, or addresses were validated and flagged for LACS^{Link} conversion. To perform LACS^{Link} processing, you must have the LACS^{Link} database installed. **N** No, do not attempt LACS^{Link} conversion. Default. **Y** Yes, attempt LACS^{Link} conversion. For a listing of the output fields returned by this option, see **LACSLink Output** on page 231
Option.PerformEWS	The Early Warning System (EWS) uses the USPS® EWS File to validate addresses that are not in the ZIP + 4® database. To perform EWS processing, you must have the EWS database installed. If an input address matches an address in the EWS file, ValidateAddress will return the following record-level result indicators: • Status="F" • Status.Code="EWSFailure" • Status.Description="Address found in EWS table" **N** No, do not perform EWS processing. Default. **Y** Yes, perform EWS processing.
Option.ExtractFirm	Specifies whether to extract the firm name from AddressLine1 through AddressLine4 and place it in the FirmName output field. This option works in cases where the input record's FirmName field is blank and there is more than one address line. **Y** Yes, extract the firm name. **N** No, do not extract the firm name. Default. To identify firm names in address lines, ValidateAddress scans the address lines for keywords and patterns that help it identify which fields are address lines and which are FirmName lines. Since this is done based on patterns, ValidateAddress may misidentify fields. The following tips can help ensure optimal firm extraction: • If possible, place the primary address elements in AddressLine1, the secondary elements in AddressLine2, Urbanization in AddressLine3, and firm in AddressLine4. If the address has no urbanization code, then place the firm name in AddressLine3 and leave AddressLine4 blank. For example,

Parameter	Description
	AddressLine1: 4200 Parliament Place **AddressLine2:** Suite 600 **AddressLine3:** Pitney Bowes Software **AddressLine4:** <blank> • When you define just two address lines, AddressLine2 is assigned to the secondary address most of the time. If you want to increase the chance that ValidateAddress will treat AddressLine2 as a firm name, put the firm name in AddressLine3 and leave AddressLine2 blank. • Numerics in a firm name (such as the "1" in "1 Stop Software") will increase the likelihood that ValidateAddress will treat the field as an address line. Here are some examples of firm name extraction: • In this example, AddressLine2 would get extracted into the FirmName output field **FirmName:** <blank> **AddressLine1:** 4200 Parliament Place Suite 600 **AddressLine2:** International Goose Feathers inc. • In this example, AddressLine3 would get extracted into the FirmName output field. **FirmName:** <blank> **AddressLine1:** 4200 Parliament Place **AddressLine2:** Suite 600 **AddressLine3:** Pitney Bowes Software • In this example, AddressLine3 would be placed in the AdditionalInputData output field. The firm name would not be extracted because the FirmName input field is not blank. **FirmName:** International Goose Feathers Inc. **AddressLine1:** 4200 Parliament Place **AddressLine2:** Suite 600 **AddressLine3:** Pitney Bowes Software • In this example, no firm name would be extracted because there is only one non-blank address line, which is always treated as the primary address element. **FirmName:** <blank> **AddressLine1:** 4200 Parliament Place Suite 600 • In this example, AddressLine2 would be treated as a secondary address element because the numeral "1" causes ValidateAddress to treat that field as a secondary address element. **FirmName:** <blank> **AddressLine1:** 4200 Parliament Place Suite 600 **AddressLine2:** Pitney Bowes Software
Option.ExtractUrb	Specifies whether to extract the urbanization name from AddressLine1 through AddressLine4 and place it in the USUrbanName output field.

Parameter	Description
	This option works in cases where the input record's USUrbanName field is blank and there is more than one address line. **Y** Yes, extract the urbanization name. **N** No, do not extract the urbanization name. Default. To identify urbanization names, ValidateAddress scans the address lines for keywords and patterns that help it identify which fields are address lines and which are urbanization name lines. Since this is done based on patterns, it is possible for ValidateAddress to incorrectly identify fields. To help ensure optimal urbanization extraction, place the primary address elements in AddressLine1, the secondary elements in AddressLine2, Urbanization in AddressLine3, and firm in AddressLine4, if possible. For example, **AddressLine1:** A1 Calle A **AddressLine2:** **AddressLine3:** URB Alamar **AddressLine4:** Pitney Bowes Software
Option.PerformSuiteLink	Specifies whether to perform Suite$^{\text{Link™}}$ processing. Suite$^{\text{Link}}$ corrects secondary address information for U.S. business addresses whose secondary address information could not be validated. If Suite$^{\text{Link}}$ processing is enabled, the firm name is matched to a database of known firm names and their secondary address information. For example, Firm Name: Pitney Bowes Software Address Line 1: 4200 Parliament Place Address Line 2: STE 1 Postal Code: 20706 In this case, Suite$^{\text{Link}}$ processing would change the suite number to the correct suite number: Firm Name: Pitney Bowes Software Address Line 1: 4200 Parliament Place Address Line 2: **STE 600** Postal Code: 20706-1844 To perform Suite$^{\text{Link™}}$ processing, you must have the Suite$^{\text{Link™}}$ database installed. This option takes one of the following values: **N** No, do not use Suite$^{\text{Link™}}$. Default. **Y** Yes, use Suite$^{\text{Link™}}$ processing. For a listing of fields returned by this option, see **SuiteLink Output** on page 233.
Option.OutputPreferredAlias	Specifies whether to use a street's preferred alias in the output. Street name aliases in the United States are alternative names given to sections of a street. There are four types of street name aliases: • **Preferred**—A preferred alias is the street name preferred locally. It typically applies only to a specific range of addresses on the street.

Parameter	Description
	• **Abbreviated**—An abbreviated alias is a variation of the street name that can be used in cases where the length of AddressLine1 is longer than 31 characters. For example, the street name 1234 BERKSHIRE VALLEY RD APT 312A could be abbreviated to 1234 BERKSHIRE VLLY RD APT 312A. • **Changed**—There has been an official street name change and the alias reflects the new name. For example if SHINGLE BROOK RD is changed to CANNING DR, then CANNING DR would be a changed alias type. • **Other**—The street alias is made up of other names for the street or common abbreviations of the street. The non-alias version of the street name is called the base street name. If the preferred alias is used in the input then the preferred alias will be the street name in the output regardless of whether you enable this option. This is one of three options that control how ValidateAddress handles street name aliases. The other two are Option.OutputStreetNameAlias and Option.OutputAbbreviatedAlias. In most cases, if you select both Option.OutputPreferredAlias and Option.OutputAbbreviatedAlias, and ValidateAddress finds both a preferred and an abbreviated alias in the postal database, the abbreviated alias will be used in the output. The exception to this rule is if the input street name is a preferred alias. In this case, the preferred alias will be used in the output. **Y** Yes, perform preferred alias processing. **N** No, do not perform preferred alias processing. Default. **Note:** If the input address contains a street name alias of type "changed" the output address will always contain the base street name regardless of the options you specify.
Option.OutputAbbreviatedAlias	Specifies whether to use a street's abbreviated alias in the output if the output address line is longer than 31 characters. This is one of three options that control how ValidateAddress handles street name aliases. The other two are Option.OutputStreetNameAlias and Option.OutputPreferredAlias. **Note:** If a preferred alias is specified in the input, the output street name will always be the preferred alias, even if you enable abbreviated street name alias processing. **Y** Yes, perform abbreviated alias processing. **N** No, do not perform abbreviated alias processing. Default. **Note:** If the input address contains a street name alias of type "changed" the output address will always contain the base street name regardless of the options you specify.
Option.DPVDetermineNoStat	Determines the "no stat" status of an address. An address is considered "no stat" if it exists but cannot receive mail, and therefore is not counted as a delivery statistic on a carrier's route (hence the term "no stat").

Parameter	Description
	Examples include buildings under construction or those that the letter carrier has identified as not likely to receive mail. **N** No, do not determine "no stat" status. Default. **Y** Yes, determine "no stat" status. **Note:** You must enable DPV processing to use this option. The result is returned in the DPVNoStat field. For more information see **LACSLink Output** on page 231
Option.DPVDetermineVacancy	Determines if the location has been unoccupied for at least 90 days. **N** No, do not determine vacancy. Default. **Y** Yes, determine vacancy. **Note:** You must enable DPV processing to use this option. The result is returned in the DPVVacant field. For more information see **LACSLink Output** on page 231
Option.ReturnVerimove	Returns VeriMove detail data in output. **N** No, do not return VeriMove detail data. Default. **Y** Yes, return VeriMove detail data.
Option.StreetMatchingStrictness	Specifies the algorithm to use when determining if an input address matches an address in the postal database. One of the following: **E** The input street name must match the database exactly. **T** The matching algorithm is "tight." **M** The matching algorithm is "medium" (default). **L** The matching algorithm is "loose."
Option.FirmMatchingStrictness	Specifies the algorithm to use when determining if an input address matches an address in the postal database. One of the following: **E** The input firm name must match the database exactly. **T** The matching algorithm is "tight." **M** The matching algorithm is "medium" (default). **L** The matching algorithm is "loose."
Option.DirectionalMatchingStrictness	Specifies the algorithm to use when determining if an input address matches an address in the postal database. One of the following: **E** The input directionals, such as the "N" in 123 N Main St., must match the database exactly. **T** The matching algorithm is "tight." **M** The matching algorithm is "medium". Default. **L** The matching algorithm is "loose."
Option.DPVSuccessfulStatusCondition	Select the match condition where a DPV result does NOT cause a record to fail.

Parameter	Description
	F Full match **P** Partial match **A** Always. Default. **Note:** You must enable DPV processing to use this option.
Option.FailOnCMRAMatch	Treat Commercial Mail Receiving Agency (CMRA) matches as failures? **N** No, do not treat CMRA matches as failures. Default. **Y** Yes, treat CMRA matches as failures. **Note:** You must enable DPV processing to use this option.
Option.StandardAddressPMBLine	Specifies where ValidateAddress places Private Mailbox (PMB) information. **N** Do not include the PMB information in Standard Address output (default). **1** Place the PMB information in AddressLine1. If you specify 1, you must set Option.StandardAddressFormat to either C or D. **2** Place the PMB information in AddressLine2.

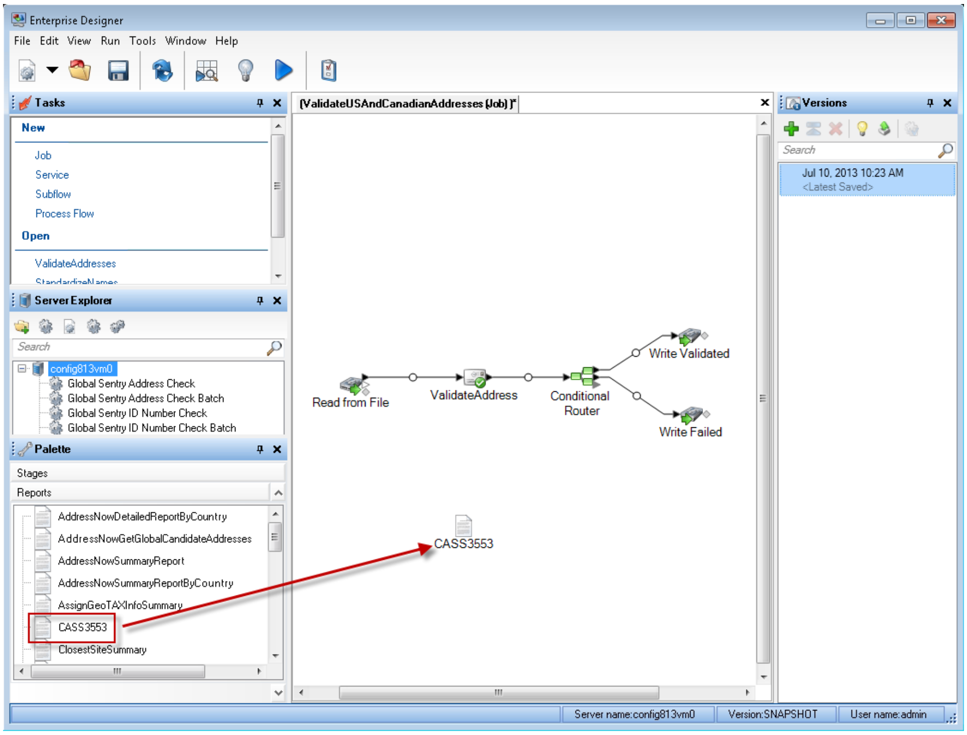
CASS Certified Processing

CASS Certified™ processing also generates the USPS CASS Detailed Report, which contains some of the same information as the 3553 report but provides much greater detail about DPV, LACS, and SuiteLink statistics. The USPS CASS Detailed Report is not required for postal discounts and does not need to be submitted with your mailing.

1. Validate Address must be in CASS Certified™ mode. If **(Not CASS Certified)** appears at the top of the window, click the **Enable CASS** button. The **Enforce CASS rules** check box will appear.
2. Click **Configure CASS 3553**. The **CASS Report Fields** dialog box appears.
3. Type the **List Processor** company name, **List Name or ID#**, and the **Number of Lists** being processed for this job.
4. Type the **Mailer Name, Address**, and **City, State, ZIP**.
5. Click **OK**.

   The List information will appear in Section B and the Mailer information in Section D of the generated USPS® CASS Form 3553.

6. In Enterprise Designer, drag the **CASS3553** report from the Reports pallet to the canvas.



7. Double-click the **CASS3553** icon on the canvas.
8. On the **Stages** tab, check the **Validate Address** checkbox. Note that if you have renamed the Validate Address stage to something else, you should check the box with the name you have given the address validation stage.
9. On the **Parameters** tab, select the format for the report. You can create the report in PDF, HTML, or plain text format.
10. Click **OK**.
11. Repeat steps 6-10 for **CASSDetail** if you want to produce the CASS Detail Report.

Canadian Address Options

Parameter	Description
Option.PerformCanadianProcessing	Specifies whether to process Canadian addresses. If you enable Canadian address processing ValidateAddress will attempt to validate Canadian addresses. If you disable Canadian address processing, Canadian addresses will fail, meaning they is returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for Canadian address processing you must disable Canadian address processing in order for your jobs to complete successfully, regardless of whether or not they contain Canadian addresses. **Note:** You must have a valid license for Canadian address processing to successfully process Canadian addresses. If you enable Canadian address processing but are not licensed for this feature, or your license has expired, you will receive an error.

Parameter	Description
	N No, do not process Canadian addresses. **Y** Yes, process Canadian addresses (default).
Option.Database.Canada	Specifies which database you want to use for validating Canadian addresses. To specify a database for Canadian address validation, select a database in the **Database** drop-down list. Only databases that have been defined in the **CAN Database Resources** panel in the Management Console are available.
Option.CanFrenchFormat	Specifies how to determine the language (English or French) to use to format the address and directional. The following example shows an address formatted in English and French: English: 123 Main St W French: 123 Rue Main O The parameter controls the formatting of the address. It also affects the spelling of the directional but not spelling of the suffix. **C** Use the street suffix returned by the matching process to determine the language. The street suffix returned by the matching process, which is used internally by ValidateAddress during processing, may be different from that in the input address. Ambiguous records are formatted like the input. Default. All addresses in Quebec are formatted using French. **S** Use the Canadian database to determine the language. The Canadian database contains data from the Canada Post Corporation (CPC). All addresses in Quebec are formatted using French. **T** Use the CanLanguage input field to determine the language. If there is a non-blank value in this field the address are formatted using French.
Option.CanFrenchFormat	For English addresses, specifies the default apartment label to use in the output if there is no apartment label in the input address. This setting is ignored if you specify Option.CanStandardAddressFormat=F. **Apt** Use "Apt" as the label. Default. **Apartment** Use "Apartment" as the label. **Suite** Use "Suite" as the label. **Unit** Use "Unit" as the label.

Parameter	Description
Option.CanFrenchApartmentLabel	For French addresses, specifies the default apartment label to use in the output if there is no apartment label in the input address. This setting is ignored if you specify Option.CanStandardAddressFormat=F. **App** Use "App" as the label. Default. **Appartement** Use "Appartement" as the label. **Bureau** Use "Bureau" as the label. **Suite** Use "Suite" as the label. **Unite** Use "Unite" as the label.
Option.ForceCorrectionLVR	Changes the civic and/or suite information to match the Large Volume Receiver (LVR) or single-single record (used when there is only one record for that postal code/street name/street type). **N** Do not change the civic and/or suite information to match the LVR or single-single record. The LVR record will be marked as a valid but non-correctable record (VN). The single-single record will be corrected, if possible, or processed as a non-correctable record.. **Y** Change the civic and/or suite information to match the LVR or single-single record. **Note:** If you check this box, the Statement of Address Accuracy will not be printed because this is **not** a SERP-recognized setting.
Option.CanPreferHouseNum	In cases where the house number and postal code are both valid but in conflict, you can force the postal code to be corrected based on the house number by specifying Option.CanPreferHouseNum=Y. If you do not select this option the house number is changed to match the postal code. **N** Change the house number to match the postal code. Default. **Y** Change the postal code to match the house number.
Option.CanOutputCityAlias	Specifies whether or not to return the city alias when the alias is in the input address. This option is disabled when you specify Option.CanOutputCityFormat=D. **Y** Output the city alias when the city alias is in the input. Default.

Parameter	Description
	N Never output the city alias even if it is in the input.
Option.CanNonCivicFormat	Specifies whether or not non-civic keywords are abbreviated in the output. For example, Post Office Box vs. PO Box. **A** Abbreviate non-civic keywords. Default. **F** Do not abbreviate non-civic keywords. The full keyword is used.
Option.EnableSERP	Specifies whether or not to use SERP options. **Y** Enable SERP options. **N** Do not enable SERP options. Default.
Option.CanStandardAddressFormat	Specifies where to place secondary address information in the output address. Secondary address information refers to apartment numbers, suite numbers, and similar designators. **D** Place apartment information in the location specified in the. Default. **B** Place apartment information at the at the end of the AddressLine1 field. **F** Place the apartment number only (no label) at the beginning of the AddressLine1 field. For example, 400-123 Rue Main **E** Place the apartment number and label at the beginning of the AddressLine1 field. For example, Apt 400 123 Rue Main **S** Place apartment information on a separate line. **S** Place apartment information in the same location as the input address.
Option.CanOutputCityFormat	Specifies whether to use the long, medium, or short version of the city if the city has a long name. For example, Long: BUFFALO HEAD PRAIRIE Medium: BUFFALO-HEAD-PR Short: BUFFALO-HD-PR **D** Use the default option specified by the Option.OutputShortCityName parameter. Default. If you specify Option.OutputShortCityName=V, the city is formatted as if you select for this option (see below) andY for **CanOutputCityAlias**. **S** Output short city name. **L** Output the long city name.

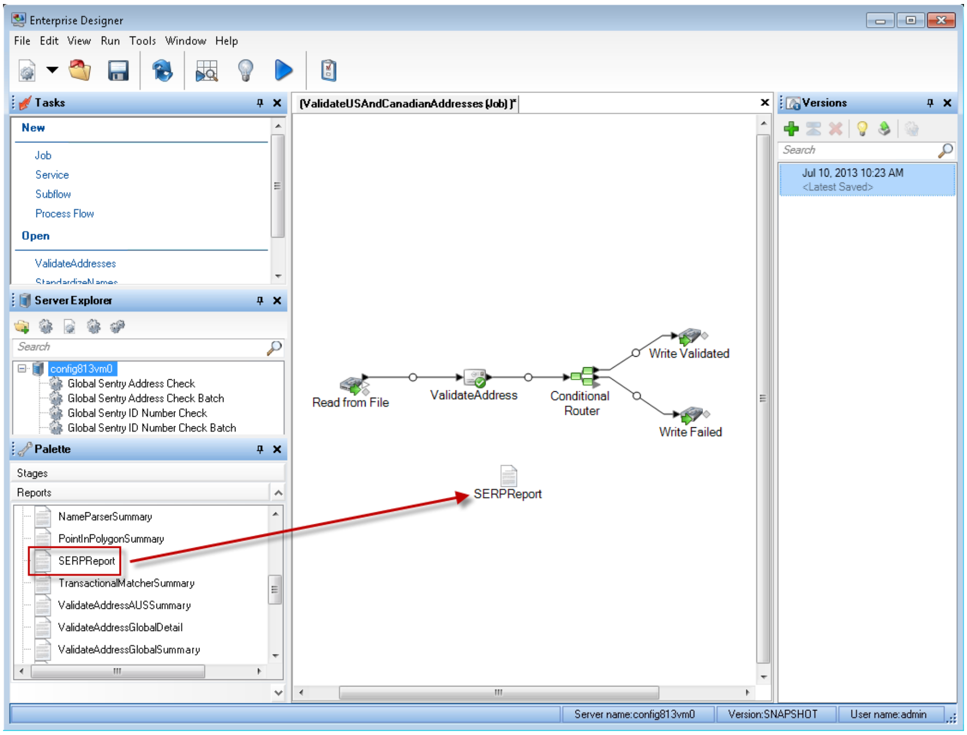
Parameter	Description
	M Output the medium city name. **I** Use the same city format as used in the input address. Output is L, M, or S.
Option.CanRuralRouteFormat	Specifies where to place rural route delivery information. An example of an address with rural route delivery information is: 36 GRANT RD RR 3 ANTIGONISH NS In this address, "RR 3" is the rural route delivery information. **A** Place rural route delivery information on the same line as the address, after the address information. Default. For example, 36 GRANT RD RR 3 **S** Place rural route delivery information on a separate address line. For example, 36 GRANT RD RR 3
Option.CanDeliveryOfficeFormat	Specifies where to place station information. An example of an address with station information is: PO BOX 8625 STN A ST. JOHN'S NL **I** Place station information in the same location as it is in the input address. Default. **A** Place station information on the same line as the address, after the address information. For example, PO BOX 8625 STN A **S** Place station information on a separate address line. For example, PO BOX 8625 STN A
Option.CanDualAddressLogic	Specifies whether ValidateAddress should return a street match or a PO Box/non-civic match when the address contains both civic and non-civic information. One of the following: **D** Use DualAddressLogic Global Option. Default. **P** Match to PO Box or other non-street data. **S** Match to street.

Parameter	Description
	For example, given the following input address: AddressLine1: 36 GRANT RD AddressLine2: RR 4 City: ANTIGONISH StateProvince: NS ValidateAddress would return one of the following: • If Option.CanDualAddressLogic is set to S, ValidateAddress returns the following: AddressLine1: 36 GRANT RD AddressLine2: RR 3 City: ANTIGONISH StateProvince: NS PostalCode: B2G 2L1 • If Option.CanDualAddressLogic is set to P, ValidateAddress returns the following: AddressLine1: RR 4 City: ANTIGONISH StateProvince: NS PostalCode: B2G 2L2 The address data that is not used to standardize the address is returned in the **AdditionalInputData** field. For more information on this option, see **Output Data Options** on page 184.

SERP Processing

1. Validate Address must be in SERP Certified™ mode. If **(Not SERP Certified)** appears at the top of the window, click the **Enable SERP settings** button. The **Configure SERP** box will appear.
2. Click **Configure SERP**. The **SERP Report Fields** dialog box appears.
3. Type your merchant **CPC number**.
4. Type the mailer **Name, Address**, and **City, State, ZIP**.
5. Click **OK**.

6. In Enterprise Designer, drag the SERPReport from the Reports pallet to the canvas.



7. Double-click the **SERPReport** icon on the canvas.
8. On the **Stages** tab, ensure that the **Validate Address** checkbox is checked. Note that if you have renamed the Validate Address stage to something else, you should check the box with the name you have given the address validation stage.
9. On the **Parameters** tab, select the format for the report. You can create the report in PDF, HTML, or plain text format. PDF format is the default.
10. Click **OK**.

Obtaining SERP Return Codes

SERP return codes indicate the quality of the input address as determined by the Canada Post's Software Evaluation and Recognition Program regulations.

To obtain SERP return codes, specify Option.OutputRecordType=P. For more information on Option.OutputRecordType, see **Output Data Options** on page 184.

SERP return codes are provided in the following output field.

Table 93: SERP Return Code Output

Response Element	Description
CanadianSERPCode	Validation/correction return code (Canadian addresses only): **V** The input was valid. Canada Post defines a "valid" address as an address that meets all the following requirements: **Note:** There are exceptions. For further information, contact the CPC. • The address must contain all required components as found in CPC's Postal Code Data Files. • The address must provide an exact match on all components for only one address in CPC's Postal Code Data Files, allowing

Response Element	Description
	for acceptable alternate words and names listed in the CPC Postal Code Data Files. • Address components must be in a form that allows recognition without ambiguity. Certain components may require "qualifiers" to identify them. For instance, a Route Service address requires the key words "Rural Route" or "RR" for differentiation from a "Suburban Service" or "SS" address with the same number. **I** The input was invalid. An "invalid" address is one that does not meet CPC requirements for a valid address (see above). Examples of this include address components that are missing, invalid, or inconsistent. **C** The input was correctable. A "correctable" address is one that can be corrected to match one, and only one, address. **N** The input was non-correctable. A "non-correctable" address is one that could be corrected a number of different ways such that ValidateAddress cannot identify a single correct version. **F** The input address was foreign (outside of Canada).

International Address Options

Addresses outside of the U.S. and Canada are referred to as "international" addresses. The following options control international address processing:

Parameter	Description
Option.PerformInternationalProcessing	Specifies whether to process international addresses (addresses outside the U.S. and Canada). If you enable international address processing ValidateAddress will attempt to validate international addresses. If you disable international address processing, international addresses will fail, meaning they is returned with an "F" in the Status output field. The output field Status.Code will say "DisabledCoder." If you are not licensed for international address processing you must disable international address processing in order for your jobs to complete successfully, regardless of whether or not they contain international addresses. **Note:** You must have a valid license for international address processing to successfully process international addresses. If you enable international address processing but are not licensed for this feature, or your license has expired, you will receive an error. **N** No, do not process international addresses.

Parameter	Description
	Y Yes, process international addresses (default).
Option.Database.International	Specifies which database you want to use for validating international addresses. To specify a database for international address validation, select a database in the **Database** drop-down list. Only databases that have been defined in the **INTL Database Resources** panel in the Management Console are available.
Option.InternationalCityStreetSearching	By default, ValidateAddress provides a balance of good address matching accuracy with good performance. If you are willing to trade matching accuracy for faster performance, use the Option.InternationalCityStreetSearching parameter to increase processing speed. When you do this, some accuracy is lost. This option only controls performance for addresses outside the U.S. and Canada. This setting affects a small percentage of records, mostly addresses in the U.K. There is no performance control for U.S. and Canadian address processing. If you use GetCandidateAddresses, the candidate addresses returned by GetCandidateAddresses may differ from the multiple matches returned by ValidateAddress if you set the performance tuning option for international addresses to any value other than 100. To control performance, specify a value from 0 to 100. A setting of 100 maximizes accuracy while a setting of 0 maximizes speed. The default is 100.
Option.AddressLineSearchOnFail	This option enables ValidateAddress to search the AddressLine input fields for the city, state/province, postal code, and country when the address cannot be matched using the values in the City, StateProvince, and PostalCode input fields. Consider enabling this option if your input addresses have the city, state/province, and postal code information in the AddressLine fields. Consider disabling this option if your input addresses use the City, State/Province and PostalCode fields. If you enable this option and these fields are used, there is an increased possibility that ValidateAddress will fail to correct values in these fields (for example a misspelled city name). **N** No, do not search the AddressLine fields. **Y** Yes, search the address line fields. Default.

Response

The output from ValidateAddress contains different information depending on the output categories you select.

Standard Address Output

Standard address output consists of four lines of the address which correspond to how the address would appear on an actual address label. City, state/province, postal code, and other data is also included in standard address output. ValidateAddress returns standard address output for validated addresses if you set Option.OutputRecordType=A. Standard address fields are always returned for addresses that could not be validated. For non-validated addresses, the standard address output fields contain the address as it appeared in the input ("pass through" data). If you want ValidateAddress to standardize address according to postal authority standards when validation fails, specify Option.OutputFormattedOnFail=Y in your request.

Table 94: Standard Address Output

Response Element	Description
AdditionalInputData	Input data not used by ValidateAddress. For more information, see **About AdditionalInputData** on page 235.
AddressLine1	If the address was validated, the first line of the validated and standardized address. If the address could not be validated, the first line of the input address without any changes.
AddressLine2	If the address was validated, the second line of the validated and standardized address. If the address could not be validated, the second line of the input address without any changes.
AddressLine3	If the address was validated, the third line of the validated and standardized address. If the address could not be validated, the third line of the input address without any changes.
AddressLine4	If the address was validated, the fourth line of the validated and standardized address. If the address could not be validated, the fourth line of the input address without any changes.
AddressLine5	For U.K. addresses only. If the address was validated, the fifth line of the validated and standardized address. If the address could not be validated, the fifth line of the input address without any changes.
City	The validated city name.
Country	The country in the format determined by what you selected in: • ISO Code • UPU Code • English • French • German • Spanish
DepartmentName	For U.K. addresses only, a subdivision of a firm. For example, Engineering Department.
FirmName	The validated firm or company name.

Response Element	Description
PostalCode	The validated ZIP Code™ or postal code.
PostalCode.AddOn	The 4-digit add-on part of the ZIP Code™. For example, in the ZIP Code™ 60655-1844, 1844 is the 4-digit add-on. (U.S. addresses only.)
PostalCode.Base	The 5-digit ZIP Code™; for example 20706 (U.S. addresses only).
StateProvince	The validated state or province abbreviation.
USUrbanName	The validated urbanization name. (U.S. addresses only.)

Parsed Address Elements Output

Output addresses are formatted in the parsed address format if you set Option.OutputRecordType=E. If you want ValidateAddress to return formatted data in the Parsed Address format when validation fails (that is, a normalized address), specify Option.OutputFormattedOnFail=Y.

Note: If you want ValidateAddress to always return parsed input data regardless of whether or not validation is successful, specify Option.OutputRecordType=I. For more information, see **Parsed Input** on page 213.

Table 95: Parsed Address Output

Response Element	Description
AdditionalInputData	Input data not used by ValidateAddress. For more information, see **About AdditionalInputData** on page 235.
ApartmentLabel	Apartment designator (such as STE or APT), for example: 123 E Main St **APT** 3
ApartmentLabel2	Secondary apartment designator, for example: 123 E Main St APT 3, 4th **Floor** **Note:** In this release, this field will always be blank.
ApartmentNumber	Apartment number. For example: 123 E Main St APT **3**
ApartmentNumber2	Secondary apartment number. For example: 123 E Main St APT 3, **4th** Floor **Note:** In this release, this field will always be blank.
CanadianDeliveryInstallationAreaName	Delivery installation name (Canadian addresses only)
CanadianDeliveryInstallationQualifierName	Delivery installation qualifier (Canadian addresses only)
CanadianDeliveryInstallationType	Delivery installation type (Canadian addresses only)
City	Validated city name

Response Element	Description
Country	Country. Format is determined by what you selected in : • ISO Code • UPU Code • English • French • German • Spanish
DepartmentName	For U.K. addresses only, a subdivision of a firm. For example, Engineering Department.
FirmName	The validated firm or company name
HouseNumber	House number, for example: **123** E Main St Apt 3
LeadingDirectional	Leading directional, for example: 123 **E** Main St Apt 3
POBox	Post office box number. If the address is a rural route address, the rural route box number will appear here.
PostalCode	Validated postal code. For U.S. addresses, this is the ZIP Code.
PrivateMailbox	Private mailbox indicator.
PrivateMailbox.Type	The type of private mailbox. Possible values include: • Standard • Non-Standard **Note:** This replaces PrivateMailboxType (no period in field name). Please modify your API calls accordingly.
RRHC	Rural Route/Highway Contract indicator
StateProvince	Validated state or province name
StreetName	Street name, for example: 123 E **Main** St Apt 3
StreetSuffix	Street suffix, for example: 123 E Main **St** Apt 3
TrailingDirectional	Trailing directional, for example: 123 Pennsylvania Ave **NW**
USUrbanName	USPS® urbanization name. Puerto Rican addresses only.

Parsed Input

The output can include the input address in parsed form. This type of output is referred to as "parsed input." Parsed input fields contain the address data that was used as input regardless of whether or not

ValidateAddress validated the address. Parsed input is different from the "parsed address elements" output in that parsed address elements contain the validated address if the address could be validated, and, optionally, the input address if the address could not be validated. Parsed input always contains the input address regardless of whether or not ValidateAddress validated the address.

To include parsed input fields in the output, set Option.OutputRecordType=I.

Table 96: Parsed Input

Response Element	Description
ApartmentLabel.Input	Apartment designator (such as STE or APT), for example: 123 E Main St **APT** 3
ApartmentNumber.Input	Apartment number, for example: 123 E Main St APT **3**
CanadianDeliveryInstallationAreaName.Input	Delivery installation name (Canadian addresses only)
CanadianDeliveryInstallationQualifierName.Input	Delivery installation qualifier (Canadian addresses only)
CanadianDeliveryInstallationType.Input	Delivery installation type (Canadian addresses only)
City.Input	Validated city name
Country.Input	Country. Format is determined by what you selected in: • ISO Code • UPU Code • English • French • German • Spanish
FirmName.Input	The validated firm or company name
HouseNumber.Input	House number, for example: **123** E Main St Apt 3
LeadingDirectional.Input	Leading directional, for example: 123 **E** Main St Apt 3
POBox.Input	Post office box number. If the address is a rural route address, the rural route box number will appear here.
PostalCode.Input	Validated postal code. For U.S. addresses, this is the ZIP Code.
PrivateMailbox.Input	Private mailbox indicator
PrivateMailbox.Type.Input	The type of private mailbox. Possible values include: • Standard • Non-Standard
RRHC.Input	Rural Route/Highway Contract indicator

Response Element	Description
StateProvince.Input	Validated state or province name
StreetName.Input	Street name, for example: 123 E **Main St** Apt 3
StreetSuffix.Input	Street suffix, for example: 123 E Main St Apt 3
TrailingDirectional.Input	Trailing directional, for example: 123 Pennsylvania Ave **NW**
USUrbanName.Input	USPS® urbanization name

Postal Data Output

If Option.OutputRecordType contains P then the following fields are returned in the output.

Table 97: Postal Data Output

Response Element	Description
CanadianSERPCode	Validation/correction return code (Canadian addresses only). For more information, see **Obtaining SERP Return Codes** on page 208.
IntHexaviaCode	For addresses in France only, a numeric code that represents the street. For information about Hexavia codes, see **www.laposte.fr**.
IntINSEECode	For addresses in France only, a numeric code that represents the city. For a listing of INSEE codes, see **www.insee.fr**.
PostalBarCode	The two-digit delivery point portion of the delivery point barcode (U.S. addresses only) For more information, see **Creating Delivery Point Barcodes** on page 188.
USAltAddr	Indicates whether or not alternate address matching logic was used, and if so which logic was used (U.S. addresses only). One of the following: **null** No alternate address scheme used. **D** Delivery point alternate logic was used. **E** Enhanced highrise alternate match logic was used. **S** Small town default logic was used. **U** Unique ZIP Code logic was used.
USBCCheckDigit	Check-digit portion of the 11-digit delivery point barcode (U.S. addresses only). For more information, see **Creating Delivery Point Barcodes** on page 188.
USCarrierRouteCode	Carrier route code (U.S. addresses only). For more information, see **Obtaining Carrier Route Codes** on page 188.
USCongressionalDistrict	Congressional district (U.S. addresses only). For more information, see **Obtaining Congressional Districts** on page 187.
USCountyName	County name (U.S. addresses only). For more information, see **Obtaining County Names** on page 188.

Response Element	Description
USFinanceNumber	The finance number in which the address resides (U.S. addresses only). The finance number is a number assigned by the USPS to an area that covers multiple ZIP Codes. ValidateAddress will successfully validate an address only if its finance number matches the finance number of the candidate address in the U.S. Database.
USFIPSCountyNumber	FIPS (Federal Information Processing Standards) county number (U.S. addresses only). For more information, see **Obtaining FIPS County Numbers** on page 188.
USLACS	Indicates whether or not the address is a candidate for LACS^{Link} conversion (U.S. addresses only). One of the following: **Y** Yes, the address is a candidate for LACS^{Link} processing. If LACS^{Link} is enabled, ValidateAddress will attempt to convert the address using the LACS^{Link} database. If the conversion attempt is successful, the output address is the new address obtained from the LACS^{Link} database. If the attempt is not successful, the address will not be converted. **N** No, the address is not a candidate for LACS^{Link} processing. LACS^{Link} processing may still be attempted if LACS^{Link} processing is requested, the LACS^{Link} database is installed, and one of the following is true: • The address matches to a Rural Route address and the RecordType.Default field returns a Y. • The input address could not be matched to any address in the U.S. Postal Database (Failures due to multiple matches are not LACS^{Link} candidates.)
USLastLineNumber	A six-character alphanumeric value that groups together ZIP Codes that share the same primary city. For example, addresses with the following two last lines would have the same last line number: Chantilly VA 20151 Chantilly VA 20152

Result Indicators

Result indicators provide information about the kinds of processing performed on an address. There are two types of result indicators:

- **Record-Level Result Indicators** on page 216
- **Field-Level Result Indicators** on page 220

Record-Level Result Indicators

Record-level result indicators provide data about the results of ValidateAddress processing for each record, such as the success or failure of the match attempt, which coder processed the address, and other details. The following table lists the record-level result indicators returned by ValidateAddress.

Table 98: Record Level Indicators

Response Element	Description
AddressFormat	The type of address data being returned: **F** French format (for example: 123 Rue Main) **E** English format (for example: 123 Main St)
Confidence	The level of confidence assigned to the address being returned. Range is from zero (0) to 100; zero indicates failure, 100 indicates a very high level of confidence that the match results are correct. For multiple matches, the confidence level is 0. For details about how this number is calculated, see **Introduction to the Validate Address Confidence Algorithm** on page 312.
CouldNotValidate	If no match was found, which address component could not be validated: • ApartmentNumber • HouseNumber • StreetName • PostalCode • City • Directional • StreetSuffix • Firm • POBoxNumber • RuralRoute **Note:** More than one component may be returned, in a comma-separated list.
CountryLevel	The category of address matching available. This is always "A" for U.S. and Canadian addresses. One of the following: **A** The address is in a country for which there is highly detailed postal data available. Addresses in this match level can have the following address elements validated and corrected, and added if missing from the input: • Postal code • City name • State/county name • Street address elements • Country name. **B** The address is in a country for which there is a medium level of postal data available. Addresses in this match level can have the following address elements validated and corrected, and added if missing from the input: • Postal code • City name • State/county name • Country name

Response Element	Description
	C The address is in a country for which the postal data is least detailed. Addresses in this match level can have the following actions performed on them: • Validate and correct country name (cannot supply missing country name) • Validate the format of the postal code (cannot supply missing postal code or validate the code)
MatchScore	MatchScore provides an indication of the degree to which the output address is correct. It is significantly different from Confidence in that Confidence is indicates how much the input address changed to obtain a match, whereas the meaning of Match Score varies between U.S. and non-U.S. addresses. For U.S. addresses, MatchScore is a one-digit score on a scale of 0 to 9 that reflects the closeness of the street-name match (after transformations by ValidateAddress, if any). Zero indicates an exact match and 9 indicates the least likely match. If no match was found, this field is blank. For non-U.S. and non-Canadian addresses, MatchScore is a five-digit score, with a maximum value of 00999. Higher numbers indicates a closer match. This field does not apply to Canadian addresses. Note that you cannot equate match scores from U.S. addresses with those of non-U.S. addresses. For example, a match score of 4 for a U.S address does not indicate the same level of match as a 00004 for a non-U.S. address. **Note:** The Validate Address and Advanced Matching Module components both use the MatchScore field. The MatchScore field value in the output of a dataflow is determined by the last stage to modify the value before it is sent to an output stage. If you have a dataflow that contains Validate Address and Advanced Matching Module components and you want to see the MatchScore field output for each stage, use a Transformer stage to copy the MatchScore value to another field. For example, Validate Address produces an output field called MatchScore and then a Transformer stage copies the MatchScore field from Validate Address to a field called AddressMatchScore. When the matcher stage runs it populates the MatchScore field with the value from the matcher and passes through the AddressMatchScore value from Validate Address.
MultimatchCount	If multiple matches were found, indicates the number of records that are possible matches.
MultipleMatches	Indicates which address component had multiple matches, if multiple matches were found: • Firm • LeadingDirectional • PostalCode • StreetName • StreetSuffix • TrailingDirectional • Urbanization

Response Element	Description
	Note: More than one component may be returned, in a comma-separated list.
ProcessedBy	Which address coder processed the address: **USA** U.S. address coder **CAN** Canadian address coder **INT** International address coder
RecordType	Type of address record, as defined by U.S. and Canadian postal authorities (supported for U.S. and Canadian addresses only): • FirmRecord • GeneralDelivery • HighRise • PostOfficeBox • RRHighwayContract • Normal
RecordType.Default	Code indicating the "default" match: **Y** The address matches a default record. **null** The address does not match a default record.
Status	Reports the success or failure of the match attempt. For multiple matches, this field is "F" for all the possible matches. **null** Success **F** Failure
Status.Code	Reason for failure, if there is one. For multiple matches, all possible matches is "MultipleMatchesFound." • DisabledCoder • InsufficientInputData • MultipleMatchesFound • UnableToValidate
Status.Description	Description of the problem, if there is one. **Possible Multiple Addresses Found** This value will appear if Status.Code=MultipleMatchesFound. **Address Not Found** This value will appear if Status.Code=UnableToValidate. **PerformUSProcessing disabled** This value will appear if Status.Code=DisabledCoder. **PerformCanadianProcessing disabled** This value will appear if Status.Code=DisabledCoder. **PerformInternationalProcessing disabled** This value will appear if Status.Code=DisabledCoder.

Field-Level Result Indicators

Field-level result indicators describe how ValidateAddress handled each address element. Field-level result indicators are returned in the qualifier "Result". For example, the field-level result indicator for HouseNumber is contained in **HouseNumber.Result**.

To enable field-level result indicators, . For more information on this option, see **Output Data Options** on page 184.

The following table lists the field-level result indicators. If a particular field does not apply to an address, the result indicator may be blank.

Table 99: Field-Level Result Indicators

Response Element	Description
AddressRecord.Result	These result codes apply to international addresses only. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input.
ApartmentLabel.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only. **C** Corrected. U.S. and Canadian addresses only. **D** Dropped. The field provided on input was removed. U.S. and Canadian addresses only. For more information, see **About AdditionalInputData** on page 235. **F** Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. U.S. and Canadian addresses only. **R** The apartment label is required but is missing from the input address. U.S. addresses only. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. Does not apply to Canadian addresses.

Response Element	Description
	V Validated. The data was confirmed correct and remained unchanged from input.
ApartmentNumber.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About AdditionalInputData** on page 235.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. U.S. addresses that are an EWS match will have a value of P. U.S. and Canadian addresses only.
	R The apartment number is required but is missing from the input address. U.S. addresses only.
	S Standardized. This option includes any standard abbreviations. Does not apply to U.S. addresses.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input.
City.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. U.S. and Canadian addresses only.
	F Hyphens missing or punctuation errors. Canadian addresses only.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. Does not apply to U.S. or Canadian addresses.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output.
	R The city is required but is missing from the input address. U.S. addresses only.

Response Element	Description
	S Standardized. This option includes any standard abbreviations. Does not apply to U.S. addresses. **U** Unmatched. Does not apply to Canadian addresses. **V** Validated. The data was confirmed correct and remained unchanged from input.
Country.Result	These result codes do not apply to U.S. or Canadian addresses. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input.
FirmName.Result	**C** Corrected. U.S. addresses only. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. U.S. and Canadian addresses only. **U** Unmatched. U.S. and Canadian addresses only. **V** Validated. The data was confirmed correct and remained unchanged from input. U.S. addresses only.
HouseNumber.Result	**A** Appended. The field was added to a blank input field. Canadian addresses only. **C** Corrected. Canadian addresses only. **D** Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About AdditionalInputData** on page 235. **F** Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses. **O** Out of range. Does not apply to U.S. or Canadian addresses. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only.

Response Element	Description
	R The house number is required but is missing from the input address. Canadian addresses only. **S** Standardized. This option includes any standard abbreviations. Does not apply to U.S. or Canadian addresses. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input.
LeadingDirectional.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only. **C** Corrected. Non-blank input was corrected to a non-blank value. U.S. addresses only. **D** Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About AdditionalInputData** on page 235. **F** Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input. Does not apply to Canadian addresses.
POBox.Result	**A** Appended. The field was added to a blank input field. Canadian addresses only. **C** Corrected. Canadian addresses only. **D** Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About AdditionalInputData** on page 235. **F** Formatted. The spacing and/or punctuation was changed to conform to postal

Response Element	Description
	standards. Does not apply to U.S. or Canadian addresses. **M** Multiple matches. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only. **R** The P.O. Box number is required but is missing from the input address. U.S. addresses only. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input.
PostalCode.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only. **C** Corrected. U.S. and Canadian addresses only. **F** Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. Does not apply to Canadian addresses. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. Does not apply to U.S. addresses. **R** The postal code is required but is missing from the input address. U.S. addresses only. **S** Standardized. This option includes any standard abbreviations. Does not apply to U.S. or Canadian addresses. **U** Unmatched. For example, if the street name does not match the postal code, both StreetName.Result and PostalCode.Result will contain U.

Response Element	Description
	V Validated. The data was confirmed correct and remained unchanged from input.
PostalCodeCity.Result	These result codes apply to international addresses only. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input.
PostalCode.Source	These result codes apply to U.S. addresses only. **FinanceNumber** The ZIP Code™ in the input was verified by using USPS® Finance Number groupings. **ZIPMOVE** The ZIP Code™ in the input address was corrected because the USPS® redrew ZIP Code™ boundaries and the address is now in a different ZIP Code™.
PostalCode.Type	**P** The ZIP Code™ contains only PO Box addresses. U.S. addresses only. **U** The ZIP Code™ is a unique ZIP Code™ assigned to a specific company or location. U.S. addresses only. **M** The ZIP Code™ is for military addresses. U.S. addresses only. **null** The ZIP Code™ is a standard ZIP Code™.
RRHC.Result	**C** Corrected. Canadian addresses only. **D** Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About AdditionalInputData** on page 235. **M** Multiple matches. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only.

Response Element	Description
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only. **R** The rural route/highway contract is required but is missing from the input address. U.S. addresses only. **S** Standardized. This option includes any standard abbreviations. U.S. and Canadian addresses only. **U** Unmatched. U.S. and Canadian addresses only. **V** Validated. The data was confirmed correct and remained unchanged from input. U.S. and Canadian addresses only.
RRHC.Type	These result codes apply to U.S. addresses only. **HC** The address is a Highway Contract address. **RR** The address is a Rural Route address.
StateProvince.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only. **C** Corrected. U.S. addresses only. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. Does not apply to U.S. or Canadian addresses. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. **R** The state is required but is missing from the input address. U.S. addresses only. **S** Standardized. This option includes any standard abbreviations. Does not apply to U.S. addresses. **U** Unmatched. Does not apply to Canadian addresses. **V** Validated. The data was confirmed correct and remained unchanged from input.
Street.Result	These result codes apply to international addresses only. **M** Multiple. The input address matched multiple records in the postal database, and

Response Element	Description
	each matching record has a different value in this field. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. **R** Street corrected. House number is out of range. Applies to UK and Japanese records only. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input.
StreetName.AbbreviatedAlias.Result	Indicates the result of abbreviated alias processing. One of the following: **null** No abbreviated alias processing attempted. **B** The StreetName field contains the base street name. **L** The standardized address length is less than 31 characters so the StreetName field contains the base name. **N** No abbreviated alias found. **Y** An abbreviated alias was found for input address. The StreetName field contains the abbreviated alias.
StreetName.Alias.Type	This result code applies to U.S. addresses only. **Note:** In previous releases this field was named StreetName.AliasType with no "." between "Alias" and "Type." This old name is obsolete. Please update your processes to use the new name StreetName.Alias.Type. **Abbreviated** The alias is an abbreviation of the street name. For example, HARTS-NM RD is an abbreviated alias for HARTSVILLE NEW MARLBORO RD. **Changed** There has been an official street name change and the alias reflects the new name. For example if SHINGLE BROOK RD is changed to CANNING DR, then CANNING DR would be a changed alias type.

Response Element	Description
	Other The street alias is made up of other names for the street or common abbreviations of the street. **Preferred** The street alias is the locally preferred alias. For example, a street is named "South Shore Dr." because it runs along the southern shore of a lake, not because it is south of a municipal demarcation line. So, "South" is not a predirectional in this case and should not be shorted to "S". So, "South Shore Dr." would be the preferred alias.
StreetName.PreferredAlias.Result	Indicates the result of preferred alias processing. One of the following: **null** No preferred alias processing attempted. **A** Preferred alias processing was not attempted because the input address matched to an alias. Preferred alias processing is only attempted for base addresses. **N** No preferred alias found. **Y** A preferred alias was found for the input address. The StreetName field contains the preferred alias.
StreetName.Result	**A** Appended. The field was added to a blank input field. Canadian addresses only. **C** Corrected. U.S. and Canadian addresses only. **D** Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About AdditionalInputData** on page 235. **F** Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. Does not apply to U.S. addresses.

Response Element	Description
	S Standardized. This option includes any standard abbreviations. U.S. and Canadian addresses only.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input.
StreetSuffix.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. U.S. and Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. and Canadian addresses only. For more information, see **About AdditionalInputData** on page 235.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only.
	S Standardized. This option includes any standard abbreviations.
	U Unmatched. Does not apply to U.S. addresses.
	V Validated. The data was confirmed correct and remained unchanged from input.
TrailingDirectional.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. U.S. and Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. and Canadian addresses only. For more information, see **About AdditionalInputData** on page 235.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.

Response Element	Description
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only. **S** Standardized. This option includes any standard abbreviations. **U** Unmatched. Does not apply to Canadian addresses. **V** Validated. The data was confirmed correct and remained unchanged from input.
USUrbanName.Result	These result codes apply to U.S. addresses only. **A** Appended. The field was added to a blank input field. **C** Corrected. **M** Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. **U** Unmatched. **V** Validated. The data was confirmed correct and remained unchanged from input.

Output from Options

ValidateAddress returns additional data depending on the options you select. For information on the output generated by each option, see the options listed in the following sections:

Enhanced Line of Travel Output

Enhanced Line of Travel processing produces the following output.

Response Element	Description
USLOTCode	Line of Travel sequence code and an indicator denoting USPS® LOT sequence. This field is in the format nnnnY where: **nnnn** The four-digit LOT code. **Y** One of the following: • **A**—Ascending LOT sequence • **D**—Descending LOT sequence

Response Element	Description
USLOTHex	A hexadecimal value that allows you to sort your file in ascending order only. The hexadecimal values range from 0 to FF ascending, then FF through 0 descending.
USLOTSequence	A two-byte value used for final sortation in place of the DPC add-on. It consists of an uppercase letter followed by a digit 0 through 9. Values range from A0 (99 descending) through J9 (00 descending), and K0 (00 ascending) through T9 (99 ascending).

LACS^{Link} Output

Response Element	Description
USLACS	Indicates whether or not the address is a candidate for LACS^{Link} conversion (U.S. addresses only). One of the following: **Y** Yes, the address is a candidate for LACS^{Link} processing. If LACS^{Link} is enabled, ValidateAddress will attempt to convert the address using the LACS^{Link} database. If the conversion attempt is successful, the output address is the new address obtained from the LACS^{Link} database. If the attempt is not successful, the address will not be converted. **N** No, the address is not a candidate for LACS^{Link} processing. LACS^{Link} processing may still be attempted if LACS^{Link} processing is requested, the LACS^{Link} database is installed, and one of the following is true: • The address matches to a Rural Route address and the RecordType.Default field returns a Y. • The input address could not be matched to any address in the U.S. Postal Database (Failures due to multiple matches are not LACS^{Link} candidates.)
USLACS.ReturnCode	Indicates the success or failure of LACS^{Link} processing. (U.S. addresses only.) **A** LACS^{Link} processing successful. Record matched through LACS^{Link} processing. **00** LACS^{Link} processing failed. No matching record found during LACS^{Link} processing. **09** LACS^{Link} processing matched the input address to an older highrise default address. The address has been converted. Rather than provide an imprecise address, LACS^{Link} processing does not provide a new address. **14** LACS^{Link} processing failed. Match found during LACS^{Link} processing but conversion did not occur due to other USPS® regulations. **92** LACS^{Link} processing successful. Record matched through LACS^{Link} processing. Unit number dropped on input.

Response Element	Description
	null LACS^{Link} did not process the record, or LACS^{Link} processing was not attempted.

RDI Output

Response Element	Description
RDI	Return values indicating address type. **B** The address is a business address. **R** The address is a residential address. **M** The address is both a residential and a business address. **null** Not checked because the address did not code at a ZIP + 4® level, or RDI™ was not performed.

DPV and CMRA Output

Response Element	Description
DPV	Indicates the results of Delivery Point Validation (DPV) processing. **Y** DPV confirmed. Mail can be delivered to the address. **N** Mail cannot be delivered to the address. **S** The building number was validated but the unit number could not be confirmed. A building number is the primary address number for a building. A unit number is a number of a distinct mailing address within a building such as an apartment, suite, floor, and so on. For example, in this address 424 is the building number and 12 is the unit number: 424 Washington Blvd. Apt. 12 Oak Park IL 60302 USA **D** The building number was validated but the unit number was missing from input. A building number is the primary address number for a building. A unit number is a number of a distinct mailing address within a building such as an apartment, suite, floor, and so on. For example, in this address 424 is the building number and 12 is the unit number: 424 Washington Blvd. Apt. 12 Oak Park IL 60302 USA **M** The address matches multiple valid delivery points. **U** The address could not be confirmed because the address did not code at the ZIP + 4® level. **V** The address caused a false-positive violation.

Response Element	Description
CMRA	Indicates if the address is a Commercial Mail Receiving Agency (CMRA) **Y** Yes, the address is a CMRA. **N** No, the address is not a CMRA. **U** Unconfirmed.
DPVFootnote	DPV footnote codes. **AA** Input address matched to the ZIP + 4® file. **A1** Input address not matched to the ZIP + 4® file. **BB** Input address matched to DPV (all components). **CC** Input address primary number matched to DPV but secondary number not match (present but not valid). **N1** Input address primary number matched to DPV but high rise address missing secondary number. **M1** Input address primary number missing. **M3** Input address primary number invalid. **P1** Input address missing PO, RR or HC Box number. **RR** Input address matched to CMRA. **R1** Input address matched to CMRA but secondary number not present.
DPVVacant	Indicates whether the building is vacant (unoccupied for 90 days). One of the following: **Y** Yes, the building is vacant. **N** No, the building is not vacant. **null** The option was not turned on.
DPVNoStat	Indicates whether the building is a "no stat" building and therefore unable to receive mail. One of the following: **Y** Yes, the building is a "no stat" building, which means the building is not receiving mail. **N** No, the building is not a "no stat" building, which means the building does receive mail. **null** The option was not turned on.

Suite$^{\text{Link}}$ Output

Response Element	Description
SuiteLinkReturnCode	Indicates whether or not ValidateAddress corrected the secondary address information (U.S. addresses only). One of the following: **A** ValidateAddress corrected the secondary address information.

Response Element	Description
	00 ValidateAddress did not correct the secondary address information. **null** Suite^{Link} was not performed. **XX** Suite^{Link} processing encountered an error. For example, an error would occur if the Suite^{Link} database is expired.
SuiteLinkMatchCode	Provides additional information on the Suite^{Link} match attempt. (U.S. addresses only) **A** ValidateAddress corrected the secondary address information. **B** ValidateAddress did not correct the secondary address information. No additional detail about the match attempt is available. **C** The words in the FirmName field are all "noise" words. Noise words are defined by the USPS® and are ignored when attempting to mach the firm name. Examples of noise words are "company" and "corporation". ValidateAddress is not able to correct secondary address information for firm names that consist entirely of noise words. For example "Company and Corporation" is all noise words. **D** The address is not a high-rise default address. Suite^{Link} matching is only done for high-rise default addresses. A high-rise default is a default to use when the address does not contain valid secondary information (the apartment number or apartment type is missing). **E** Suite^{Link} processing failed because the Suite^{Link} database is expired. **null** Suite^{Link} was not performed or there was an error.
SuiteLinkFidelity	Indicates how well ValidateAddress matched the firm name to the firm names in the Suite^{Link} database. **1** The firm name matches the Suite^{Link} database exactly. **2** Good match. All words in the firm name except one matched the firm name in the Suite^{Link} database. **3** Poor match. More than one word in the firm name did not match the firm name in the Suite^{Link} database. **null** Suite^{Link} could not match the firm name, or was not performed, or there was an error.

VeriMove Output

Response Element	Description
VeriMoveDataBlock	Indicates whether or not ValidateAddress should return a 250-byte field containing input data to pass to VeriMove Express. This field contains the Detail Results Indicator data required by VeriMove. For more

Response Element	Description
	information about the contents of this field, see the VeriMove User's Guide. One of the following: **Y** Yes, return the field VeriMoveDataBlock.. **N** No, do not return the field VeriMoveDataBlock.

About AdditionalInputData

ValidateAddress ignores some input data during the address standardization process. This extraneous data (sometimes referred to as "dropped data") is returned in the AdditionalInputData. Some examples of dropped data include:

- Delivery instructions (for example, "Leave at back door")
- Phone numbers (for example, "555-135-8792")
- Attention lines (for example, "Attn: John Smith")

Data such as this is generally not embedded in an address. If it is embedded, ValidateAddress can usually identify this extraneous data and return it in the AdditionalInputData.

Note: ValidateAddress does not return dropped data from split indicia addresses. A split indicia address is one where a primary address is split between multiple address lines. For example, if the primary address is "1 Green River Valley Rd" then the following would be a split indicia version of this address: 1 Green River Valley Rd 01230

If there is more than one piece of dropped data in an address, each piece of data is separated by a semicolon and a space ("; ") for U.S. addresses and a space for addresses outside the U.S. The order of dropped data in AdditionalInputData is:

1. Care of, mail stop (U.S. addresses only)
2. Other extraneous data found on address lines
3. Entire unused data lines

For example, if this is the input address:

123 Main St C/O John Smith
Apt 5 Drop at back dock
jsmith@example.com
555-123-4567
05674

Then AdditionalInputData would contain:

C/O John Smith; Apt 5 Drop At Back Dock; 555-123-4567; Jsmith@example.com; 555-123-4567

ValidateAddress can handle the following types of extraneous data:

- **Care Of Data** on page 235
- **Extraneous Data on Its Own Address Line** on page 236
- **Extraneous Data Within an Address Line** on page 236
- **Dual Addresses** on page 237

Care Of Data

For U.S. addresses only, "care of" data is returned in AdditionalInputData. The following addresses contain examples of "care of" data:

123 Main St C/O John Smith
Apt 5
05674

123 Main St
Apt 5 ATTN John Smith
05674

123 Main St Apt 5
MailStop 2
05674

Extraneous Data on Its Own Address Line

ValidateAddress returns extraneous data on its own address line for U.S. and Canadian addresses.

For U.S. addresses, ValidateAddress uses the first two non-blank address lines to perform address standardization, unless either the firm name extraction or urbanization code extraction options are enabled (see **Address Line Processing for U.S. Addresses** on page 183 for more information). Data on other address lines is returned in AdditionalInputData. In the following address, "John Smith" would be returned in AdditionalInputData because it is in the third non-blank address line and ValidateAddress only uses the first two non-blank address lines for U.S. addresses.

123 Main St
Apt 5
John Smith
05674

If one of either of the first two non-blank address lines contains extraneous data, that data is returned in AdditionalInputData. For example, in the following addresses "John Smith" would be returned in AdditionalAddressData.

123 Main St
John Smith
05674

John Smith
123 Main St
05674

In the following address both "John Smith" and "Apt 5" would both be returned in AdditionalInputData. "John Smith" would be returned because it is extraneous data in one of the first two address lines and "Apt 5" would be returned because U.S. address data must be in the first two non-blank address lines.

John Smith
123 Main St
Apt 5
05674

Extraneous Data Within an Address Line

Extraneous data that is within an address line is returned in AdditionalInputData. For example, in the following addresses "John Smith" would be returned in AdditionalInputData.

123 Main St John Smith
05674

123 Main St Apt 5 John Smith
05674

123 Main St John Smith
Apt 5
05674

123 Main St
Apt 5 John Smith
05674

For U.S. addresses, only extraneous data at the end of the address line is returned in AdditionalInputData. Extraneous data that is not at the end of an address line is not returned for U.S. addresses. For example, in the following addresses "John Smith" is not returned.

John Smith 123 Main St
05674

123 Main John Smith St
05674

The AdditionalInputData will sometimes contain the original street name or suffix if the street name was changed to obtain a match and the street name or suffix was at the end of a line. For example this address:

Pitney Bowes Software
4200 Parlament
Lanham MD

ValidateAddress would correct the spelling of the street name and add the suffix, returning "4200 Parliament Pl" as the corrected street address and "Parlament" in AdditionalInputData.

Dual Addresses

A dual address is an address that contains both street and PO Box/Rural Route/Highway Contract information. Depending on the processing options you select, the portion of the dual address that is not used for address standardization may be returned in AdditionalInputData. For more information, see **About Dual Address Logic** on page 193.

ValidateAddressGlobal

ValidateAddressGlobal provides enhanced address standardization and validation for addresses outside the U.S. and Canada. ValidateAddressGlobal can also validate addresses in the U.S. and Canada but its strength is validation of addresses in other countries. If you process a significant number of addresses outside the U.S. and Canada, you should consider using ValidateAddressGlobal.

ValidateAddressGlobal is part of the Universal Addressing Module.

ValidateAddressGlobal performs several steps to achieve a quality address, including transliteration, parsing, validation, and formatting.

Character Set Mapping and Transliteration

ValidateAddressGlobal handles international strings and their complexities. It uses fully Unicode enabled string processing which enables the transliteration of non-roman characters into the Latin character set and mapping between different character sets.

Character set mapping and transliteration features include:

- Support for over 30 different character sets including UTF-8, ISO 8859-1, GBK, BIG5, JIS, EBCDIC
- Proper "elimination" of diacritics according to language rules
- Transliteration for various alphabets into Latin Script
- Greek (BGN/PCGN 1962, ISO 843 - 1997)
- Cyrillic (BGN/PCGN 1947, ISO 9 - 1995)
- Hebrew
- Japanese Katakana, Hiragana and Kanji
- Chinese Pinyin (Mandarin, Cantonese)
- Korean Hangul

Address Parsing, Formatting, and Standardization

Restructuring incorrectly fielded address data is a complex and difficult task especially when done for international addresses. People introduce many ambiguities as they enter address data into computer systems. Among the problems are misplaced elements (such as company or personal names in street address fields) or varying abbreviations that are not only language, but also country specific. ValidateAddressGlobal identifies address elements in address lines and assigns them to the proper fields. This is an important precursor to the actual validation. Without restructuring, "no match" situations might result.

Properly identified address elements are also important when addresses have to be truncated or shortened to fit specific field length requirements. With the proper information in the right fields, specific truncation rules can be applied.

- Parses and analyzes address lines and identifies individual address elements
- Processes over 30 different character sets
- Formats addresses according to the postal rules of the country of destination
- Standardizes address elements (such as changing AVENUE to AVE)

Global Address Validation

Address validation is the correction process where properly parsed address data is compared against reference databases supplied by postal organizations or other data providers. ValidateAddressGlobal validates individual address elements to check for correctness using sophisticated fuzzy matching technology and produces standardized and formatted output based on postal standards and user preferences. FastCompletion validation type can be used in quick address entry applications. It allows input of truncated data in several address fields and generates suggestions based on this input.

In some cases, it is not possible to fully validate an address. Here ValidateAddressGlobal has a unique deliverability assessment feature that classifies addresses according to their probable deliverability.

Resource URL

JSON endpoint:

```
http://server:port/rest/ValidateAddressGlobal/results.json
```

XML endpoint:

```
http://server:port/rest/ValidateAddressGlobal/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/ValidateAddressGlobal/results.json?
City.StateProvince=NY&Data.AddressLine1=1+Global+View&
Data.City=Troy&Data.Country=USA
```

The JSON returned by this request would be:

```
{"output": [{
   "Country.Input": "USA",
   "AddressLine1.Input": "1 Global View",
   "City.Input": "Troy",
   "Country": "UNITED STATES",
   "AddressLine1": "1 GLOBAL VW",
   "HouseNumber": "1",
   "StreetName": "GLOBAL",
   "StreetSuffix": "VW",
   "City": "TROY",
   "PostalCode": "12180-8371",
   "PostalCode.Base": "12180",
   "PostalCode.AddOn": "8371",
   "StateProvince": "NEW YORK",
   "County": "RENSSELAER",
   "LastLine": "TROY NY 12180-8371",
   "AddressBlock1": "1 GLOBAL VW",
   "AddressBlock2": "TROY NY 12180-8371",
   "ProcessStatus": "C4",
   "ProcessStatus.Description": "Corrected - all elements have been
checked",
   "ModeUsed": "BATCH",
   "CountOverflow": "NO",
   "MailabilityScore": "5",
   "Confidence": "82.09",
   "ElementResultStatus": "88F088E0F000000000E0",
   "ElementInputStatus": "00600050600000000060",
   "ElementRelevance": "11101010100000000010",
   "AddressType": "S",
   "AMAS.Status": "EAM0",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/ValidateAddressGlobal/results.xml?
Data.AddressLine1=1+Global+View&Data.City=Troy&
Data.Country=USA&Data.StateProvince=NY
```

The XML returned by this request would be:

```
<ns2:xml.ValidateAddressGlobalResponse
xmlns:ns2="http://www.pb.com/spectrum/services/ValidateAddressGlobal">
   <ns2:output>
      <ns2:Address>
         <ns2:Country>UNITED STATES</ns2:Country>
         <ns2:AddressLine1>1 GLOBAL VW</ns2:AddressLine1>
         <ns2:HouseNumber>1</ns2:HouseNumber>
         <ns2:StreetName>GLOBAL</ns2:StreetName>
         <ns2:StreetSuffix>VW</ns2:StreetSuffix>
         <ns2:City>TROY</ns2:City>
         <ns2:PostalCode>12180-8371</ns2:PostalCode>
         <ns2:PostalCode.Base>12180</ns2:PostalCode.Base>
         <ns2:PostalCode.AddOn>8371</ns2:PostalCode.AddOn>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:County>RENSSELAER</ns2:County>
         <ns2:LastLine>TROY NY 12180-8371</ns2:LastLine>
         <ns2:AddressBlock1>1 GLOBAL VW</ns2:AddressBlock1>
         <ns2:AddressBlock2>TROY NY 12180-8371</ns2:AddressBlock2>
         <ns2:ProcessStatus>C4</ns2:ProcessStatus>
         <ns2:ProcessStatus.Description>
              Corrected - all elements have been checked
         </ns2:ProcessStatus.Description>
         <ns2:ModeUsed>BATCH</ns2:ModeUsed>
         <ns2:CountOverflow>NO</ns2:CountOverflow>
         <ns2:MailabilityScore>5</ns2:MailabilityScore>
         <ns2:Confidence>85.09</ns2:Confidence>
         <ns2:ElementResultStatus>
              88F0F8E0F000000000E0
         </ns2:ElementResultStatus>
         <ns2:ElementInputStatus>
              00606050600000000060
         </ns2:ElementInputStatus>
         <ns2:ElementRelevance>
              11101010100000000010
         </ns2:ElementRelevance>
         <ns2:AddressType>S</ns2:AddressType>
         <ns2:AMAS.Status>EAM0</ns2:AMAS.Status>
```

```
            <ns2:user_fields/>
         </ns2:Address>
      </ns2:output>
</ns2:xml.ValidateAddressGlobalResponse>
```

Request

Parameters for Input Data

ValidateAddressGlobal takes a standard address as input. All addresses use this format no matter what country the address is from.

Table 100: ValidateAddressGlobal Input

Parameter	Format	Description
Data.AddressLine1 through Data.AddressLine6	String [79]	These fields contain address line data. AddressLine1 contains the first address line, AddressLine2 contains the second address line, and so forth. Note that the city, state/province, and postal code information should be placed in their respective fields, not address line fields. For example: **AddressLine1:** 17413 Blodgett Road **AddressLine2:** PO Box 123 **City:** Mount Vernon **StateProvice:** WA **PostalCode:** 97273 **Country:** USA If the input address is not already parsed into the appropriate address line and City, StateProvice, and PostalCode fields, use the UnformattedLine fields instead of the address line fields.
Data.City	String [79]	City name
Data.StateProvince	String [79]	State or province.
Data.PostalCode	String [79]: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999	The postal code for the address. In the U.S. this is the ZIP Code®.
Data.Contact	String [79]	The name of the addressee. For example, "Mr. Jones".
Data.Country	String [79]	The country name. If no value is specified in the or option, you must specify a country.
Data.FirmName	String [79]	Company or firm name

Parameter	Format	Description
Data.Street	String [79]	Street
Data.Number	Building [79]	Number
Data.Building	String [79]	Building
Data.SubBuilding	String [79]	SubBuilding
Data.DeliveryService	String [79]	DeliveryService
Data.UnformattedLine1 through Data.UnformattedLine10	String [79]	Use these fields if the input address is completely unparsed and you want ValidateAddressGlobal to attempt to parse the address into the appropriate fields. For example: **UnformattedLine1:** 17413 Blodgett Road **UnformattedLine2:** PO Box 123 **UnformattedLine3:** Mount Vernon WA 97273 **UnformattedLine4:** USA This address would be parsed into these output fields: **AddressLine1:** 17413 Blodgett Road **AddressLine2:** PO Box 123 **City:** Mount Vernon **StateProvice:** WA **PostalCode:** 97273 **Country:** USA **Note:** If you specify input in the unformatted line fields you must specify the entire address using only unformatted line fields. Do not use other fields such as City or StateProvince in combination with unformatted line fields.

Parameters for Options

Input Options

Table 101: ValidateAddressGlobal Input Options

Parameter	Description/Valid Values
Option.Database.AddressGlobal	Specifies the database resource containing the postal data to use for address validation. Only databases that have been defined in the **Global Database Resources** panel in the Management Console are available. For more information, see the *Spectrum™ Technology Platform Administration Guide*.
Option.Input.DefaultCountryISO3	Specifies a default country to use when the input record does not contain explicit country information. Specify the country using the ISO3 country code. If you do not specify a default country each input record must have the country

Parameter	Description/Valid Values
	specified in the Country input field. For a list of ISO codes see **Country ISO Codes and Module Support** on page 292.
Option.Input.ForceCountryISO3	Causes address records to be always treated as originating from the country specified here, overriding the country in the address record and the default country. Specify the country using the ISO3 country code. For a list of ISO codes, see **Country ISO Codes and Module Support** on page 292.
Option.Input.FormatDelimiter	Enables you to use non-standard formatting for multi-line addresses in input files. Acceptable values for this field include the following: • CRLF (default) • LF • CR • SEMICOLON (2101 MASSACHUSETTS AVE NW **;** WASHINGTON DC 20008) • COMMA (2101 MASSACHUSETTS AVE NW **,** WASHINGTON DC 20008) • TAB (2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008) • PIPE (2101 MASSACHUSETTS AVE NW **|** WASHINGTON DC 20008) • SPACE (2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008) **Note:** The same value must be selected for both the input option and output option.

Output Options

Table 102: ValidateAddressGlobal Output Options

Parameter	Description
Option.Result.MaximumResults	This option specifies the maximum number of candidate addresses to return. The default is 1. The maximum is 20. If you are using FastCompletion mode, you may want to enter a number greater than 1 to ensure you are provided with multiple options for completing a field.
Option.Result.IncludeInputs	Specifies whether to include the input data in the output. If enabled, the output will contain fields that end with .Input containing the corresponding input field. For example, the output field AddressLine1.Input would contain the data specified in the input field AddressLine1. **TRUE** Include the input data in the output. **FALSE** Do not include the input data in the output (default).
Option.Result.StateProvinceType	Specifies the format for the StateProvince field. One of the following. **ABBREVIATION** Return the abbreviation for the state or province. For example, North Carolina would be returned as "NC". **COUNTRY_STANDARD** Return either the abbreviation or the full name depending on the format used by the country's postal authority. (Default)

Parameter	Description	
	EXTENDED	Return the full name of the state or province, not the abbreviation. For example "North Carolina".
Option.Result.CountryType	Specifies the language or code to use for the country name returned by ValidateAddressGlobal.	
	ISO2	The two-character ISO code for the country
	ISO3	The three-character ISO code for the country
	ISO_NUMBER	The ISO country number
	NAME_CN	Chinese
	NAME_DA	Danish
	NAME_DE	German
	NAME_EN	English (default)
	NAME_ES	Spanish
	NAME_FI	Finnish
	NAME_FR	French
	NAME_GR	Greek
	NAME_HU	Hungarian
	NAME_IT	Italian
	NAME_JP	Japanese
	NAME_KR	Korean
	NAME_NL	Dutch
	NAME_PL	Polish
	NAME_PT	Portuguese
	NAME_RU	Russian
	NAME_SA	Sanskrit
	NAME_SE	Swedish
Option.Result.PreferredScript	Specifies the alphabet in which the output should be returned. The alphabet in which the data is returned differs from country to country. For most countries the output will be Latin I regardless of the selected preferred language.	
	ASCII_Extended	ASCII characters with expansion of special characters (e.g. Ã– = OE)
	ASCII_Simplified	ASCII characters
	Database	(default) Latin I or ASCII characters (as per reference database standard)
	Latin	Latin I characters
	Latin_Alt	Latin I characters (alternative transliteration)

Parameter	Description
	Postal_Admin_Alt Latin I or ASCII characters (local postal administration alternative) **Postal_Admin_Pref** Latin I or ASCII characters (as preferred by local postal administration) For countries that use an alphabet other than Latin I, the returned alphabet differs from country to country. For more information, see **Alphabets for Non-Latin 1 Countries** on page 245.
Option.Result.PreferredLanguage	Specifies the language in which the output should be returned. The alphabet in which the data is returned differs from country to country, but for most countries the output will be Latin, regardless of the selected preferred language. **DATABASE** Language derived from reference data for each address. Default. **ENGLISH** English locality and state/province names output, if available.
Option.Result.Casing	Specifies the casing of the output. **NATIVE** Output will be based on the reference database standard. **UPPER** Output will be in upper case for all countries. **LOWER** Output will be in lower case for all countries. **MIXED** Casing determined by country-specific rules. **NOCHANGE** For parse mode, returns the data the way it was entered. For validation mode, uses the casing found in the reference data and according to postal rules. Values that could not be checked against the reference data will retain their input casing.
Option.Result.FormatDelimiter	Enables you to use non-standard formatting for multi-line addresses in the output. Acceptable values for this field include the following: • CRLF (default) • LF • CR • SEMICOLON (2101 MASSACHUSETTS AVE NW **;** WASHINGTON DC 20008) • COMMA (2101 MASSACHUSETTS AVE NW **,** WASHINGTON DC 20008) • TAB (2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008) • PIPE (2101 MASSACHUSETTS AVE NW **|** WASHINGTON DC 20008) • SPACE (2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008) **Note:** The same value must be selected for both the input option and output option.

Alphabets for Non-Latin 1 Countries

For countries that use an alphabet other than Latin I, the returned alphabet differs from country to country. The following table shows how the output is returned for specific countries. All countries that are not listed use the value specified in the field option.

Country	Database			Latin			
RUS	Cyrillic	Cyrillic	Cyrillic	CYRILLIC_ISO	CYRILLIC_BGN	CYRILLIC_ISO + LATIN_SIMPLE	CYRILLIC_ISO + LATIN
JPN	Kanji	Kanji	Kana	JAPANESE	JAPANESE	JAPANESE + LATIN_SIMPLE	JAPANESE + LATIN
CHN	Hanzi	Hanzi	Hanzi	CHINESE_ MANDARIN	CHINESE_ CANTONESE	CHINESE_ MANDARIN + LATIN_SIMPLE	CHINESE_ MANDARIN + LATIN
HKG	Hanzi	Hanzi	Hanzi	CHINESE_ CANTONESE	CHINESE_ MANDARIN	CHINESE_ CANTONESE + LATIN_SIMPLE	CHINESE_ CANTONESE + LATIN
TWN	Hanzi	Hanzi	Hanzi	CHINESE_ CANTONESE	CHINESE_ MANDARIN	CHINESE_ CANTONESE + LATIN_SIMPLE	CHINESE_ CANTONESE + LATIN
GRC	Greek	Greek	Greek	GREEK_ISO	GREEK_BGN	GREEK_ISO + LATIN_SIMPLE	GREEK_ISO + LATIN
KOR	Latin	Hangul	Hanja	KOREAN	KOREAN	KOREAN + LATIN_SIMPLE	KOREAN + LATIN
ISR	Latin	Hebrew	Hebrew	HEBREW	HEBREW	HEBREW + LATIN_SIMPLE	HEBREW + LATIN
ROM	Latin-3	Latin-3	Latin-3	Latin-3	Latin-3	LATIN_SIMPLE	LATIN
POL	Latin-2	Latin-2	Latin-2	Latin-2	Latin-2	LATIN_SIMPLE	LATIN
CZE	Latin-2	Latin-2	Latin-2	Latin-2	Latin-2	LATIN_SIMPLE	LATIN
CRI	Latin-2	Latin-2	Latin-2	Latin-2	Latin-2	LATIN_SIMPLE	LATIN
HUN	Latin-2	Latin-2	Latin-2	Latin-2	Latin-2	LATIN_SIMPLE	LATIN
MDA	Latin-2	Latin-2	Latin-2	Latin-2	Latin-2	LATIN_SIMPLE	LATIN
SVK	Latin-2	Latin-2	Latin-2	Latin-2	Latin-2	LATIN_SIMPLE	LATIN
LAT	Latin-7	Latin-7	Latin-7	Latin-7	Latin-7	LATIN_SIMPLE	LATIN

Process Options

Table 103: ValidateAddressGlobal Process Options

Parameter	Description
Option.Process.OptimizationLevel	Use this option to set the appropriate balance between processing speed and quality. One of the following:

Parameter	Description
	NARROW The parser will honor input assignment strictly, with the exception of separation of House Number from Street information. **STANDARD** The parser will separate address element more actively as follows: • Province will be separated from Locality information • PostalCode will be separated from Locality information • House Number will be separated from Street information • SubBuilding will be separated from Street information • DeliveryService will be separated from Street information • SubBuilding will be separated from Building information • Locality will be separated from PostalCode information **WIDE** Parser separation will happen similarly to Standard, but additionally up to 10 parsing candidates will be passed to validation for processing. Validation will widen its search tree and take additional reference data entries into account for matching. Please note that adjusting the optimization level might have no effect for countries that lack the postal reference data information required for the kind of separation described above. Increasing separation granularity from Narrow to Standard consumes some processing power, but the major impact on processing speed is from validation processing a larger search tree, thus increasing the number of data accesses and comparisons for the optimization level Wide, in an attempt to make the most out of the input data given.
Option.Process.Mode	Specifies the type of processing to perform on the addresses. One of the following: **BATCH** Use this mode in batch processing environments when no human input or selection is possible. It is optimized for speed and will terminate its attempts to correct an address when ambiguous data is encountered that cannot be corrected automatically. The Batch processing mode will fall back to Parse mode when the database is missing for a specific country. **CERTIFIED** Use this mode in batch processing environments for Australian mail. Validate Address Global is certified by Australia Post's Address Matching Approval System (AMAS). It will standardize and validate your mail against the Postal Address File, providing postal discounts and allowing for the least amount of undeliverable pieces. **FASTCOMPLETION** Use this mode if you want to use FastCompletion mode to enter truncated data in address fields and have Validate Address Global generate suggestions. For example, if you work in a call center or point-of-sale environment, you can enter just part of an address element and the FastCompletion feature will provide valid options for the complete element.

Parameter	Description
	INTERACTIVE Use this mode when working in interactive environments to generate suggestions when an address input is ambiguous. This validation type is especially useful in data entry environments when capturing data from customers or prospects. It requires the input of an almost-complete address and will attempt to validate or correct the data provided. If ambiguities are detected, this validation type will generate up to 20 suggestions that can be used for pick lists. The Interactive processing mode will fall back to Parse mode when the respective database is missing for a specific country.
	PARSE Use this mode for separating address input into tokens for subsequent processing in other systems, bypassing validation. For example, you could use this mode when address data of already high quality simply needs to be tokenized quickly for export to an external system or for use by a downstream stage.
Option.Process.MatchingScope	Specifies how closely an address must match the reference data in order for the address to be validated. One of the following: **Note:** These settings may not have an effect for countries lacking the necessary level of detail in the postal reference data.
	ALL All address elements must match.
	DELIVERYPOINT_LEVEL Validate Global Address must achieve a match on StateProvince, PostalCode, City/Locality/Suburb, street, house number, and sub building.
	STREET_LEVEL Validate Global Address must achieve a match on StateProvince, PostalCode, City/Locality/Suburb, and street.
	LOCALITY_LEVEL Validate Global Address must achieve a match on StateProvince, PostalCode, and City/Locality/Suburb.

Response

Address Data

Table 104: Parsed Address Elements

Response Element	Description
AddressBlock1-9	The AddressBlock output fields contain a formatted version of the standardized or normalized address as it would be printed on a physical mailpiece. Validate Address Global formats the address into address blocks using postal authority standards. Each line of the address is returned in a separate address block field. There can be up to nine

Response Element	Description
	address block output fields: AddressBlock1 through AddressBlock9. For example, this input address: AddressLine1: 4200 Parliament Place AddressLine2: Suite 600 City: Lanham StateProvince: MD PostalCode: 20706 Results in this address block output: AddressBlock1: 4200 PARLIAMENT PL STE 600 AddressBlock2: LANHAM MD 20706-1882
AddressLine1-6	If the address was validated, the address line fields contain the validated and standardized address lines. If the address could not be validated, the address line fields contain the input address without any changes. Note that the last line of the address is contained in the LastLine field. For example: AddressLine1: 4200 PARLIAMENT PL STE 600 LastLine: LANHAM MD 20706-1882
AdministrativeDistrict	An area smaller than a state/province but larger than a city.
ApartmentLabel	The flat or unit type (such as STE or APT), for example: 123 E Main St **Apt** 3
ApartmentNumber	The flat or unit number, for example: 123 E Main St Apt **3**
BlockName	An estate or block name.
BuildingName	The name of a building, for example Sears Tower.
City	The name of the town or city. For example, **Vancouver**, BC.
City.AddInfo	Additional information about the city.
City.SortingCode	A code used by the postal authority to speed up delivery in certain countries for large localities, for example Prague or Dublin.
Contact	The name of the addressee. For example, **Mr. Jones**.
Country	The country in the language or code specified in the option.
County	Dependent state or province information that further subdivides a state or province. An example would be a U.S. county.
FirmName	The name of a company.
Floor	Information that further subdivides a building, e.g. the suite or apartment number. For example: 123 E Main St Apt 3, **4th Floor**
HouseNumber	The house number 1, for example: 298A-1B New South Head Rd
LastLine	Complete last address line (city, state/province, and postal code).
LeadingDirectional	Street directional that precedes the street name. For example, the N in 138 N Main Street.
Locality	Dependent place name that further subdivides a Locality. Examples are colonias in Mexico, Urbanisaciones in Spain.
POBox	Post Box descriptor (POBox, Postfach, Case Postale etc.) and number.

Response Element	Description
PostalCode	The postcode for the address. The format of the postcode varies by country.
PostalCode.AddOn	The second part of a postcode. For example, for Canadian addresses this will be the LDU. For U.S. addresses this is the ZIP + 4 add on. This field is not used by most countries.
PostalCode.Base	The base portion of the postcode.
Room	A room number in a building.
SecondaryStreet	The name of a secondary street or rural route.
StateProvince	The name of the state or province.
StreetName	The name of street where property is located, for example: 123 E **Main** St Apt 3
StreetSuffix	The street suffix, for example: 123 E Main **St** Apt 3
SubBuilding	A portion of a building, such as a suite. For example, Suite 102.
Suburb	Dependent place name that further subdivides a Locality. An example would be Mahalle in Turkey.
Territory	The name of a territory. Territories are larger than a state/province.
TrailingDirectional	The trailing directional, for example: 123 Pennsylvania Ave **NW**

Original Input Data

This option outputs the original input data in <FieldName>.Input fields.

Table 105: Original Input Data

Response Element	Format	Description
AddressLine1.Input	String [79]	First address line
AddressLine2.Input	String [79]	Second address line
AddressLine3.Input	String [79]	Third address line
AddressLine4.Input	String [79]	Fourth address line
AddressLine5.Input	String [79]	Fifth address line
AddressLine6.Input	String [79]	Sixth address line
City.Input	String [79]	City name

Response Element	Format	Description
StateProvince.Input	String [79]	State or province
PostalCode.Input	String [79]:	The postal code for the address. In the U.S. this is the ZIP Code. One of these formats: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999
Contact.Input	String [79]	The name of the addressee. For example, "Mr. Jones".
Country.Input	String [79]	Specify the country using the format you chose for input country format (English name, ISO code, or UPU code). For a list of valid values, see **Country ISO Codes and Module Support** on page 292.
FirmName.Input	String [79]	Company or firm name
Street.Input	String [79]	Street
Number.Input	Building [79]	Number
Building.Input	String [79]	Building
SubBuilding.Input	String [79]	SubBuilding
DeliveryService.Input	String [79]	DeliveryService

Result Codes

These output fields contain information about the result of the validation processing.

Table 106: Result Codes

Response Element	Result Code
AddressType	For United States and Canada addresses only, the AddressType field indicates the type of address. One of the following: **F** The address was validated/corrected to the firm name. **B** The address was validated/corrected to the building name. **G** The address is a general delivery address. **H** The address was validated/corrected to the high-rise default. **L** The address is a large volume receiver.

Response Element	Result Code
	M The address is a military address. **P** The address was validated/corrected to PO box. **R** The address was validated/corrected to a rural route. **S** The address was validated/corrected to a street address. **U** The address could not be validated/corrected so the type is unknown.
Confidence	The level of confidence assigned to the address being returned. Range is from zero (0) to 100; zero indicates failure, 100 indicates a very high level of confidence that the match results are correct.
CountOverflow	Indicates whether the number of candidate addresses exceeds the number returned. One of the following: **Yes** Yes, there are additional candidate addresses. To obtain the additional candidates, increase the value. **No** No, there are no additional candidates.
ElementInputStatus	ElementInputStatus provides per element information on the matching of input elements to reference data. The values in this field vary depending on whether you are using batch mode or parse mode. For information about the value in this field, see **Interpreting ElementInputStatus, ElementResultStatus, and ElementRelevance** on page 253.
ElementRelevance	Indicates which address elements are actually relevant from the local postal authority's point of view. For information about the value in this field, see **Interpreting ElementInputStatus, ElementResultStatus, and ElementRelevance** on page 253.
ElementResultStatus	ElementResultStatus categorizes the result in more detail than the ProcessStatus field by indicating if and how the output fields have been changed from the input fields. For information about the value in this field, see **Interpreting ElementInputStatus, ElementResultStatus, and ElementRelevance** on page 253.
MailabilityScore	An estimate of how likely it is that mail sent to the address would be successful delivered. One of the following: **5** Completely confident of deliverability **4** Almost certainly deliverable **3** Should be deliverable **2** Fair chance **1** Risky **0** No chance
ModeUsed	Indicates the processing mode used. The processing mode is specified in the option. For a description of the modes, see **Process Options** on page 245.
MultimatchCount	If the address was matched to multiple candidate addresses in the reference data, this field contains the number of candidate matches found.

Response Element	Result Code
ProcessStatus	Provides a general description of the output quality. For a more detailed description of the output quality, see the ElementResultStatus field. One of the following:
	V4 Verified. The input data is correct. All elements were checked and input matched perfectly.
	V3 Verified. The input data is correct on input but some or all elements were standardized or the input contains outdated names or exonyms.
	V2 Verified. The input data is correct but some elements could not be verified because of incomplete reference data.
	V1 Verified. The input data is correct but the user standardization has deteriorated deliverability (wrong element user standardization - for example, postcode length chosen is too short). Not set by validation.
	C4 Corrected. All elements have been checked.
	C3 Corrected, but some elements could not be checked.
	C2 Corrected, but delivery status unclear (lack of reference data).
	C1 Corrected, but delivery status unclear because user standardization was wrong. Not set by validation.
	I4 Data could not be corrected completely, but is very likely to be deliverable. Single match (e.g. HNO is wrong but only 1 HNO is found in reference data).
	I3 Data could not be corrected completely, but is very likely to be deliverable. Multiple matches (e.g. HNO is wrong but more than 1 HNO is found in reference data).
	I2 Data could not be corrected, but there is a slim chance that the address is deliverable.
	I1 Data could not be corrected and is unlikely to be delivered.
	RA Country recognized from the Force country Setting
	R9 Country recognized from DefaultCountryISO3 Setting
	R8 Country recognized from name without errors
	R7 Country recognized from name with errors
	R6 Country recognized from territory
	R5 Country recognized from province
	R4 Country recognized from major town
	R3 Country recognized from format
	R2 Country recognized from script
	R1 Country not recognized - multiple matches
	R0 Country not recognized
	S4 Parsed perfectly
	S3 Parsed with multiple results

Response Element	Result Code	
	S2	Parsed with errors. Elements change position.
	S1	Parse Error. Input Format Mismatch.
	N1	Validation Error: No validation performed because country was not recognized.
	N2	Validation Error: No validation performed because required reference database is not available.
	N3	Validation Error: No validation performed because country could not be unlocked.
	N4	Validation Error: No validation performed because reference database is corrupt or in wrong format.
	N5	Validation Error: No validation performed because reference database is too old.
	N6	Validation Error: No validation performed because input data was insufficient.
	Q3	FastCompletion Status: Suggestions are available - complete address.
	Q2	FastCompletion Status: Suggested address is complete but combined with elements from the input (added or deleted).
	Q1	FastCompletion Status: Suggested address is not complete (enter more information).
	Q0	FastCompletion Status: Insufficient information provided to generate suggestions.
Status	Reports the success or failure of the processing attempt.	
	null	Success
	F	Failure
Status.Code	The reason for the failure, if there was one.	
Status.Description	A description of the reason for the failure, if there was one.	

Interpreting ElementInputStatus, ElementResultStatus, and ElementRelevance

The ElementInputStatus, ElementResultStatus, and ElementRelevance output fields contain a series of digits that describe the outcome of the validation operation in detail. ElementInputStatus contains some information for parsing operations.

This is what an ElementInputStatus value looks like:

44606040600000000060

This is what an ElementResultStatus value looks like:

88F0F870F00000000040

This is what an ElementRelevance value looks like:

11101010100000000000

To understand the values in these fields you need to know which element each position represents, and the meaning of the values in each position. For example, the first digit indicates the result from the PostalCode.Base output field. The position meanings are listed below.

- Position 1—PostalCode.Base
- Position 2—PostalCode.AddOn
- Position 3—City
- Position 4—Locality and Suburb
- Position 5—StateProvice
- Position 6—County
- Position 7—StreetName
- Position 8—SecondaryStreet
- Position 9—HouseNumber
- Position 10—Number level 1
- Position 11—POBox
- Position 12—Delivery service level 1
- Position 13—Building level 0
- Position 14—BuildingName
- Position 15—Sub building level 0
- Position 16—Floor and Room
- Position 17—FirmName
- Position 18—Organization level 1
- Position 19—Country
- Position 20—Territory

For ElementInputStatus, the possible values for validation are:

- 0—Empty
- 1—Not found
- 2—Not checked (no reference data)
- 3—Wrong - Set by validation only: The reference database suggests that either Number or DeliveryService is out of valid number range. Input is copied, not corrected for batch mode, for interactive mode and FastCompletion suggestions are provided.
- 4—Matched with errors in this element
- 5—Matched with changes (inserts and deletes) For example:
  - Parsing: Splitting of house number for "MainSt 1"
  - Validation: Replacing input that is an exonym or dropping superfluous fielded input that is invalid according to the country reference database
- 6—Matched without errors

For ElementInputStatus, the possible values for parsing are:

- 0—Empty
- 1—Element had to be relocated
- 2—Matched but needed to be normalized
- 3—Matched

For ElementRelevance, the possible values for parsing are:

- 0—Empty
- 1—Element had to be relocated
- 2—Matched but needed to be normalized
- 3—Matched

For ElementResultStatus, the possible values are (for all address elements apart from country):

- 0—Empty
- 1—Not validated and not changed. Original is copied.
- 2—Not validated but standardized.

- 3—Validated but not changed due to invalid input, database suggests that number is out of valid ranges. Input is copied, not corrected - this status value is only set in batch mode.
- 4—Validated but not changed due to lack of reference data.
- 5—Validated but not changed due to multiple matches. Only set in batch mode, otherwise multiple suggestions that replace the input are marked as corrected (status value 7).
- 6—Validated and changed by eliminating the input value
- 7—Validated and changed due to correction based on reference data
- 8—Validated and changed by adding value based on reference data
- 9—Validated, not changed, but delivery status not clear (e.g. DPV value wrong; given number ranges that only partially match reference data).
- C—Validated, verified but changed due to outdated name
- D—Validated, verified but changed from exonym to official name
- E—Validated, verified but changed due to standardization based on casing or language. Validation only sets this status if input fully matches a language alternative.
- F—Validated, verified and not changed due to perfect match

For Country (position 19 & 20), the following values are possible:

- 0—Empty
- 1—Country not recognized
- 4—Country recognized from DefaultCountryISO3 setting
- 5—Country not recognized - multiple matches
- 6—Country recognized from script
- 7—Country recognized from format
- 8—Country recognized from major town
- 9—Country recognized from province
- C—Country recognized from territory
- D—Country recognized from name with errors
- E—Country recognized from name without errors
- F—Country recognized from ForceCountryISO3 setting

ValidateAddressLoqate

ValidateAddressLoqate standardizes and validates addresses using postal authority address data. ValidateAddress Loqate can correct information and format the address using the format preferred by the applicable postal authority. It also adds missing postal information, such as postal codes, city names, state/province names, and so on.

ValidateAddressLoqate also returns result indicators about validation attempts, such as whether or not ValidateAddressLoqate validated the address, the level of confidence in the returned address, the reason for failure if the address could not be validated, and more.

During address matching and standardization, ValidateAddressLoqate separates address lines into components and compares them to the contents of the Universal Addressing Module databases. If a match is found, the input address is *standardized* to the database information. If no database match is found, ValidateAddressLoqate optionally *formats* the input addresses. The formatting process attempts to structure the address lines according to the conventions of the appropriate postal authority.

ValidateAddressLoqate is part of the Universal Addressing Module.

Resource URL

JSON endpoint:

```
http://server:port/rest/ValidateAddressLoqate/results.json
```

XML endpoint:

```
http://server:port/rest/ValidateAddressLoqate/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/ValidateAddressLoqate/results.json?
Data.AddressLine1=1+Global+View&Data.City=Troy&Data.StateProvince=NY
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Confidence": "95",
   "CouldNotValidate": "",
   "ProcessedBy": "LOQATE",
   "MatchScore": "100.0",
   "AddressLine1": "1 Global Vw",
   "AddressLine2": "",
   "City": "Troy",
   "StateProvince": "NY",
   "PostalCode": "12180-8371",
   "Country": "United States",
   "FirmName": "",
   "PostalCode.Base": "12180",
   "PostalCode.AddOn": "8371",
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/ValidateAddressLoqate/results.xml?
Data.AddressLine1=1+Global+View&Data.City=Troy&Data.StateProvince=NY
```

The XML returned by this request would be:

```
<ns2:xml.ValidateAddressLoqateResponse
xmlns:ns2="http://www.pb.com/spectrum/services/ValidateAddressLoqate">
   <ns2:output_port>
      <ns2:Address>
         <ns2:Confidence>95</ns2:Confidence>
         <ns2:CouldNotValidate/>
         <ns2:ProcessedBy>LOQATE</ns2:ProcessedBy>
         <ns2:MatchScore>100.0</ns2:MatchScore>
         <ns2:AddressLine1>1 Global Vw</ns2:AddressLine1>
         <ns2:AddressLine2/>
         <ns2:City>Troy</ns2:City>
         <ns2:StateProvince>NY</ns2:StateProvince>
         <ns2:PostalCode>12180-8371</ns2:PostalCode>
         <ns2:PostalCode.Base>12180</ns2:PostalCode.Base>
         <ns2:PostalCode.AddOn>8371</ns2:PostalCode.AddOn>
         <ns2:Country>United States</ns2:Country>
         <ns2:FirmName/>
         <ns2:user_fields/>
      </ns2:Address>
   </ns2:output_port>
</ns2:xml.ValidateAddressLoqateResponse>
```

Request

Parameters for Input Data

ValidateAddressLoqate takes an address as input. All addresses use this format regardless of the address's country. See **Address Line Processing for U.S. Addresses** on page 257 for important information about how address line data is processed for U.S. addresses.

Table 107: Input Format

Parameter	Format	Description
Data.AddressLine1	String	The first address line.
Data.AddressLine2	String	The second address line.
Data.AddressLine3	String	The third address line.
Data.AddressLine4	String	The fourth address line.
Data.City	String	The city name.
Data.Country	String	The country code or name, in any of the following formats: • Two-character ISO 3116-1 Alpha-2 country code • Three-character ISO 3116-1 Alpha-3 country code • English country name See **Country ISO Codes and Module Support** on page 292 for a list of ISO codes.
Data.FirmName	String	The company or firm name.
Data.PostalCode	String	The postal code for the address in one of the following formats: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999
Data.StateProvince	String	The state or province.

Address Line Processing for U.S. Addresses

The input fields AddressLine1 through AddressLine4 are handled differently for U.S. addresses depending on whether the firm name extraction or urbanization code extraction options are enabled. If either of these options is enabled, ValidateAddressLoqate will look at the data in all four fields to validate the address and extract the requested data (firm name and/or urbanization code). If neither of these options is enabled, ValidateAddressLoqate uses only the first two non-blank address line fields in its validation attempt. The data in the other address line fields is returned in the output field AdditionalInputData. For example,

AddressLine1: A1 Calle A
AddressLine2:
AddressLine3: URB Alamar
AddressLine4: Pitney Bowes Software

In this address, if either firm name extraction or urbanization code extraction were enabled, ValidateAddressLoqate would examine all four address lines. If neither firm name extraction nor urbanization code extraction were enabled, ValidateAddressLoqate would examine AddressLine1 and

AddressLine3 (the first two non-blank address lines) and attempt to validate the address using that data; the data in AddressLine4 would be returned in the output field AdditionalInputData.

Options

The following table lists the options that control the type of information returned by ValidateAddressLoqate.

Table 108: Output Data Options

Parameter	Description
Option.Database.Loqate	Specifies which database you want to use for validating international addresses. To specify a database for international address validation, select a database in the **Database** drop-down list.
Option.OutputFieldLevelReturnCodes	Specifies whether to include field-level result indicators. Field-level result indicators describe how ValidateAddressLoqate handled each address element. Field-level result indicators are returned in the qualifier "Result". For example, the field-level result indicator for HouseNumber is contained in **HouseNumber.Result**. For a complete listing of result indicator output fields, see **Result Indicators** on page 266. **N** No, do not output field-level return codes (default). **Y** Yes, output field-level return codes.
Option.OutputFormattedOnFail	Specifies whether to return a formatted address when an address cannot be validated. The address is formatted using the preferred address format for the address's country. If this option is not selected, the output address fields are blank when ValidateAddressLoqate cannot validate the address. **Note:** This option applies only to U.S. and Canadian addresses. Formatted data will not be returned for any other address. **N** No, do not format failed addresses (default). **Y** Yes, format failed addresses. Formatted addresses are returned using the format specified by the**Include a standard address**, **Include address line elements**, and **Include postal information** check boxes. Note that if you select **Include address line elements**, the parsed address elements will contain the parsed, validated address for addresses that could be validated. If the address could not be validated the parsed address elements will contain the input address in parsed form. If you always want the output to contain the input address in parsed form, regardless of whether or not ValidateAddressLoqate could validate the address, select **Include standardized input address elements**. If you check this option, you must select **Include a standard address** and/or **Include address line elements**. Formatted addresses are returned using the format specified by the **OutputRecordType** option. Note that if you specify

Parameter	Description
	OutputRecordType=E, the parsed address elements will contain the parsed, validated address for addresses that could be validated. If the address could not be validated the parsed address elements will contain the input address in parsed form. If you always want the output to contain the input address in parsed form, regardless of whether or not ValidateAddressLoqate could validate the address, specify **OutputRecordType=I**. If you specify Y, you must specify "A" and/or "E" for OutputRecordType. Formatted addresses are returned using the format specified by the **Option.OutputRecordType** option. Note that if you specify **Option.OutputRecordType=E**, the parsed address elements will contain the parsed, validated address for addresses that could be validated. If the address could not be validated the parsed address elements will contain the input address in parsed form. If you always want the output to contain the input address in parsed form, regardless of whether or not ValidateAddressLoqate could validate the address, specify **Option.OutputRecordType=I**. If you specify Y, you must specify "A" and/or "E" for Option.OutputRecordType.
Option.OutputAddressBlocks	Specifies whether to return a formatted version of the address as it would be printed on a physical mailpiece. Each line of the address is returned in a separate address block field. There can be up to nine address block output fields: **AddressBlock1** through **AddressBlock9**. For example, this input address: AddressLine1: 4200 Parliament Place AddressLine2: Suite 600 City: Lanham StateProvince: MD PostalCode: 20706 Results in this address block output: AddressBlock1: 4200 PARLIAMENT PL STE 600 AddressBlock2: LANHAM MD 20706-1882 AddressBlock3: UNITED STATES OF AMERICA ValidateAddressLoqate formats the address into address blocks using postal authority standards. The country name is returned using the Universal Postal Union country name. Note that the option does not affect the country name in the address block, it only affects the name returned in the **Country** output field. For addresses outside the U.S. and Canada, if ValidateAddressLoqate is unable to validate the address, no address blocks are returned. For addresses in the U.S. and Canada, address blocks are returned even if validation fails. One of the following:

Parameter	Description
	N No, do not return address blocks. Default. **Y** Yes, return address blocks.
Option.OutputCasing	Specifies the casing of the output data. One of the following: **M** The output in mixed case (default). For example: 123 Main St Mytown FL 12345 **U** The output in upper case. For example: 123 MAIN ST MYTOWN FL 12345
Option.HomeCountry	Specifies the default country. You should specify the country where most of your addresses reside. For example, if most of the addresses you process are in Canada, specify Canada. ValidateAddressLoqate uses the country you specify to attempt validation when it cannot determine the country from the StateProvince, PostalCode, and Country address fields. The valid country names are: Afghanistan, Albania, Algeria, American Somoa, Andorra, Angola, Anguilla, Antigua And Barbuda, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, Bosnia And Herzegovina, Botswana, Brazil, British Virgin Islands, Brunei Darussalam, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Cayman Islands, Central African Republic, Chad, Chile, China, Colombia, Comoros Islands, Congo, Cook Islands, Costa Rica, Cote D'Ivoire, Croatia, Cuba, Cyprus, Czech Republic, Democratic Republic Of Congo, Denmark, Djibouti, Dominica, Dominican Republic, East Timor, Ecuador, Egypt, El Salvador, Equitorial Guinea, Eritrea, Estonia, Ethiopia, Falkland Islands, Faroe Islands, Federated States Of Micronesia, Fiji, Finland, France, French Guiana, Gabon, Gambia, Germany, Ghana, Gibraltar, Greece, Greenland, Grenada, Guadeloupe, Guam, Guatemala, Guinea, Guinea Bissau, Guyana, Haiti, Holy See, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, Israel, Italy, Jamaica, Japan, Jordan, Kazakhstan, Kenya, Kiribati, Korea, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Marshall Islands, Martinique, Mauritania, Mauritius, Mayotte, Mexico, Moldova, Monaco, Mongolia, Monserrat, Morocco, Mozambique, Myanmar, Namibia, Nauru, Nepal, Netherlands Antilles, New Caledonia, New Zealand, Nicaragua, Niger, Nigeria, Niue, Norway, Oman, Pakistan, Palau, Panama, Papua New Guinea, Paraguay, Peru, Philippines, Pitcairn Islands, Poland, Portugal, Puerto Rico, Qatar, Republic Of Georgia, Republic Of Korea, Republic Of Singapore, Reunion,

Parameter	Description
	Romania, Russia, Rwanda, Saint Helena, Saint Kitts And Nevis, Saint Lucia, Saint Pierre And Miquelon, Saint Vincent And The Grenadines, Samoa, San Marino, Sao Tome And Principe, Saudi Arabia, Senegal, Seychelles, Sierra Leone, Slovakia, Slovenia, Solomon Islands, Somalia, South Africa, Spain, Sri Lanka, Sudan, Surivalue, Swaziland, Sweden, Switzerland, Syria, Tahiti, Taiwan, Tajikistan, Tanzania, Thailand, The Netherlands, Togo, Tonga, Trinidad And Tobago, Tristan Da Cunha, Tunisia, Turkey, Turkmenistan, Turks And Caicos Islands, Tuvalu, Uganda, Ukraine, United Arab Emirates, United Kingdom, United States, Uruguay, Uzbekistan, Vanuatu, Venezuela, Vietnam, Virgin Islands (US), Wallis And Futuna, Yemen, Yugoslavia, Zambia, Zimbabwe
Option.OutputCountryFormat	Specifies the format to use for the country name returned in the **Country** output field. For example, if you select English, the country name "Deutschland" would be returned as "Germany". **E** Use English country names (default). **I** Use two-letter ISO abbreviation for the countries instead of country names. **U** Use Universal Postal Union abbreviation for the countries instead of country names.
Option.OutputScript	Specifies the alphabet or script in which the output should be returned. This option is bi-directional and generally takes place from Native to Latin and Latin to Native. **Input** Do not perform transliteration and provide output in the same script as the input (default). **Native** Output in the native script for the selected country wherever possible. **Latn** Use English values.
Option.KeepMultimatch	Indicates whether or not to return multiple address for those input addresses that have more than one possible match. **Y** Yes, return multiple matches (default). **N** No, do not return multiple matches. For more information, see **Returning Multiple Matches** on page 261.
Fail Multiple Matches	Fails multiple addresses for those input addresses that have more than one possible match.

Returning Multiple Matches

If ValidateAddressLoqate finds multiple address in the postal database that are possible matches for the input address, you can have ValidateAddressLoqate return the possible matches. For example, the following address matches multiple addresses in the U.S. postal database:

PO BOX 1 New York, NY

Options

To return multiple matches, use the options described in the following table.

Table 109: Multiple Match Option

	Description/Valid Values
	Indicates whether or not to return multiple address for those input addresses that have more than one possible match. number between 1 and 10 that indicates the maximum number of addresses to return. The default value is 1. **Note:** The difference between and is that a multiple match will return a failure if, whereas a multiple match will return one record if. To identify which output addresses are candidate addresses, you must. When you do this, records that are candidate addresses will have one or more "M" values in the field-level result indicators.

Output

When you choose to return multiple matches, the addresses are returned in the address format you specify. For information on specifying address format, see **Options** on page 258. To identify which records are the candidate addresses, look for multiple "M" values in the field-level result indicators. For more information**Result Indicators** on page 266, see .

Response

The output from ValidateAddressLoqate contains various information depending on the output categories you select.

Standard Address Output

Standard address output consists of four lines of the address which correspond to how the address would appear on an actual address label. City, state/province, postal code, and other data is also included in standard address output. ValidateAddressLoqate returns standard address output for validated addresses if you. Standard address fields are always returned for addresses that could not be validated regardless of whether or not you. For non-validated addresses, the standard address output fields contain the address as it appeared in the input ("pass through" data). If you want ValidateAddressLoqate to standardize address according to postal authority standards when validation fails,.

Table 110: Standard Address Output

Response Element	Description
AdditionalInputData	Input data that could not be matched to a particular address component. For more information, see **About Additional Input Data**.
AddressLine1-4	If the address was validated, the first line of the validated and standardized address. If the address could not be validated, the first line of the input address without any changes. There can be up to four address block output fields: AddressLine1 through AddressLine4.

Response Element	Description
City	The validated city name.
Country	The country in the format determined by what you selected in: • ISO Code • UPU Code • English
FirmName	The validated firm or company name.
PostalCode	The validated ZIP Code™ or postal code.
PostalCode.AddOn	The 4-digit add-on part of the ZIP Code™. For example, in the ZIP Code™ 60655-1844, 1844 is the 4-digit add-on.
PostalCode.Base	The 5-digit ZIP Code™; for example 20706.
StateProvince	The validated state or province abbreviation.

Parsed Address Elements Output

Output addresses are formatted in the parsed address format if you. If you want ValidateAddressLoqate to return formatted data in the Parsed Address format when validation fails (that is, a normalized address),.

Note: If you want ValidateAddressLoqate to always return parsed input data regardless of whether or not validation is successful,. For more information, see **Parsed Input** on page 264.

Table 111: Parsed Address Output

Response Element	Description
AddressBlock1-9	The AddressBlock output fields contain a formatted version of the standardized or normalized address as it would be printed on a physical mailpiece. Validate Address Global formats the address into address blocks using postal authority standards. Each line of the address is returned in a separate address block field. There can be up to nine address block output fields: AddressBlock1 through AddressBlock9. For example, this input address: AddressLine1: 4200 Parliament Place AddressLine2: Suite 600 City: Lanham StateProvince: MD PostalCode: 20706 Results in this address block output: AddressBlock1: 4200 PARLIAMENT PL STE 600 AddressBlock2: LANHAM MD 20706-1882
ApartmentLabel	Apartment designator (such as STE or APT), for example: 123 E Main St **APT** 3

Response Element	Description
ApartmentNumber	Apartment number. For example: 123 E Main St APT **3**
ApartmentNumber2	Secondary apartment number. For example: 123 E Main St APT 3, **4th** Floor **Note:** In this release, this field will always be blank.
City	Validated city name
Country	Country. Format is determined by what you selected in : • ISO Code • UPU Code • English
FirmName	The validated firm or company name
HouseNumber	House number, for example: **123** E Main St Apt 3
LeadingDirectional	Leading directional, for example: 123 **E** Main St Apt 3
POBox	Post office box number. If the address is a rural route address, the rural route box number will appear here.
PostalCode	Validated postal code. For U.S. addresses, this is the ZIP Code.
StateProvince	Validated state or province name
StreetName	Street name, for example: 123 E **Main** St Apt 3
StreetSuffix	Street suffix, for example: 123 E Main **St** Apt 3
TrailingDirectional	Trailing directional, for example: 123 Pennsylvania Ave **NW**

Parsed Input

The output can include the input address in parsed form. This type of output is referred to as "parsed input." Parsed input fields contain the address data that was used as input regardless of whether or not ValidateAddress validated the address. Parsed input is different from the "parsed address elements" output in that parsed address elements contain the validated address if the address could be validated, and, optionally, the input address if the address could not be validated. Parsed input always contains the input address regardless of whether or not ValidateAddress validated the address.

To include parsed input fields in the output,.

Table 112: Parsed Input

Response Element	Description
ApartmentLabel.Input	Apartment designator (such as STE or APT), for example: 123 E Main St **APT** 3
ApartmentNumber.Input	Apartment number, for example: 123 E Main St APT **3**
City.Input	Validated city name
Country.Input	Country. Format is determined by what you selected in: • ISO Code • UPU Code • English
FirmName.Input	The validated firm or company name
HouseNumber.Input	House number, for example: **123** E Main St Apt 3
LeadingDirectional.Input	Leading directional, for example: 123 **E** Main St Apt 3
POBox.Input	Post office box number. If the address is a rural route address, the rural route box number will appear here.
PostalCode.Input	Validated postal code. For U.S. addresses, this is the ZIP Code.
StateProvince.Input	Validated state or province name
StreetName.Input	Street name, for example: 123 E **Main St** Apt 3
StreetSuffix.Input	Street suffix, for example: 123 E Main St Apt 3
TrailingDirectional.Input	Trailing directional, for example: 123 Pennsylvania Ave **NW**

Geocode Output

ValidateAddressLoqate returns the latitude/longitude, geocoding match code, dependent and double dependent localities, dependent thoroughfare, subadministrative and superadministrative areas, and the search distance as output. Match codes describe how well the geocoder matched the input address to a known address; they also describe the overall status of a match attempt. Search distance codes represent how close the geocode is to the actual physical location of an address. he output returned is in the DataTable class. For information on the DataTable class, see the "API Fundamentals" section .

Table 113: Standard Address Output

Response Element	Description
Geocode.MatchCode	This two-byte code reflects the status and level of geocode matching for an address. The first byte represents the geocoding status and is one of the following:

<table>
<tr><th>Response Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td><b>A</b></td><td>Multiple candidate geocodes were found to match the input address, and an average of these was returned</td></tr>
<tr><td><b>I</b></td><td>A geocode was able to be interpolated from the input addresses location in a range</td></tr>
<tr><td><b>P</b></td><td>A single geocode was found matching the input address</td></tr>
<tr><td><b>U</b></td><td>A geocode was not able to be generated for the input address</td></tr>
</table>
The second byte represents the level of geocoding matching and is one of the following:
<table>
<tr><td><b>5</b></td><td>Delivery point (post box or subbuilding)</td></tr>
<tr><td><b>4</b></td><td>Premise or building</td></tr>
<tr><td><b>3</b></td><td>Thoroughfare</td></tr>
<tr><td><b>2</b></td><td>Locality</td></tr>
<tr><td><b>1</b></td><td>Administrative area</td></tr>
<tr><td><b>0</b></td><td>None</td></tr>
</table>
<b>Table 114: City/Street/Postal Code Centroid Match Codes</b>
<table>
<tr><th>Element</th><th>Match Code</th></tr>
<tr><td>Address Point</td><td>P4</td></tr>
<tr><td>Address Point Interpolated</td><td>I4</td></tr>
<tr><td>Street Centroid</td><td>A4/P3</td></tr>
<tr><td>Postal Code/City Centroid</td><td>A3/P2/A2</td></tr>
</table>
<b>Note:</b> Geocode.Match.Code does not return two coordinates for a street segment (such as the beginning and ending of a portion of a street). Instead, with input resulting in return codes of I3 (interpolated to thoroughfare or street level, where no input premise number was provided), the complete street is used in the computation.
</td></tr>
<tr><td>Latitude</td><td>Eight-digit number in degrees and calculated to five decimal places (in the format specified).</td></tr>
<tr><td>Longitude</td><td>Eight-digit number in degrees and calculated to five decimal places (in the format specified).</td></tr>
<tr><td>SearchDistance</td><td>The radius of accuracy in meters, providing an indication of the probable maximum distance between the given geocode and the actual physical location. This field is derived from and dependent upon the accuracy and coverage of the underlying reference data.</td></tr>
</table>

Result Indicators

Result indicators provide information about the kinds of processing performed on an address. There are two types of result indicators:

- Record-Level Result Indicators
- Field-Level Result Indicators

Record-Level Result Indicators

Record-level result indicators provide data about the results of ValidateAddressLoqate processing for each record, such as the success or failure of the match attempt, which coder processed the address, and other details. The following table lists the record-level result indicators returned by ValidateAddressLoqate.

Table 115: Record Level Indicators

Response Element	Description
Confidence	The level of confidence assigned to the address being returned. Range is from zero (0) to 100; zero indicates failure, 100 indicates a very high level of confidence that the match results are correct. For multiple matches, the confidence level is 0. For details about how this number is calculated, see **Introduction to the Validate Address Confidence Algorithm** on page 312.
CouldNotValidate	If no match was found, which address component could not be validated: • ApartmentNumber • HouseNumber • StreetName • PostalCode • City • Directional • StreetSuffix • Firm • POBoxNumber **Note:** More than one component may be returned, in a comma-separated list.
MatchScore	MatchScore provides an indication of the similarity between the input data and the closest reference data match. It is significantly different from Confidence in that Confidence indicates how much the input address changed to obtain a match, whereas the meaning of Match Score varies between U.S. and non-U.S. addresses. The int getFieldMatchscore (unit record, const char*) field is a decimal value between 0 and 100 that reflects the similarity between the identified input data and the closest reference data match. A result of 100 indicates that no changes other than alias, casing, or diacritic changes have been made to the input data. A result of 0 indicates that there is no similarity between the input data and closest reference data match. **Note:** The Validate Address Loqate and Advanced Matching Module components both use the MatchScore field. The MatchScore field value in the output of a dataflow is determined by the last stage to modify the value before it is sent to an output stage. If you have a dataflow that contains Validate Address Loqate and Advanced Matching Module components and you want to see the MatchScore field output for each stage, use a Transformer stage to copy the MatchScore value to another field. For

Response Element	Description
	example, Validate Address Loqate produces an output field called MatchScore and then a Transformer stage copies the MatchScore field from Validate Address Loqate to a field called AddressMatchScore. When the matcher stage runs it populates the MatchScore field with the value from the matcher and passes through the AddressMatchScore value from Validate Address Loqate.
ProcessedBy	Which address coder processed the address: **LOQATE** The Loqate coder processed the address.
Status	Reports the success or failure of the match attempt. For multiple matches, this field is "F" for all the possible matches. **null** Success **F** Failure
Status.Code	Reason for failure, if there is one. • UnableToValidate
Status.Description	Description of the problem, if there is one. **Address Not Found** This value will appear if Status.Code=UnableToValidate.

Field-Level Result Indicators

Field-level result indicators describe how ValidateAddressLoqate handled each address element. Field-level result indicators are returned in the qualifier "Result". For example, the field-level result indicator for HouseNumber is contained in **HouseNumber.Result**.

To enable field-level result indicators, .

The following table lists the field-level result indicators. If a particular field does not apply to an address, the result indicator may be blank.

Table 116: Field-Level Result Indicators

Response Element	Description
ApartmentLabel.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only. **C** Corrected. U.S. and Canadian addresses only. **D** Dropped. The field provided on input was removed. U.S. and Canadian addresses only. For more information, see **About Additional Input Data**. **F** Formatted. The spacing and/or punctuation was changed to conform to postal standards. **P** Pass-through. The data was not used in the validation process, but it was preserved in the output. U.S. and Canadian addresses only.

Response Element	Description	
	R	The apartment label is required but is missing from the input address. U.S. addresses only.
	S	Standardized. This option includes any standard abbreviations.
	U	Unmatched. Does not apply to Canadian addresses.
	V	Validated. The data was confirmed correct and remained unchanged from input.
ApartmentNumber.Result	**A**	Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C	Corrected. Canadian addresses only.
	D	Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About Additional Input Data**.
	F	Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	P	Pass-through. The data was not used in the validation process, but it was preserved in the output. U.S. addresses that are an EWS match will have a value of P. U.S. and Canadian addresses only.
	R	The apartment number is required but is missing from the input address. U.S. addresses only.
	S	Standardized. This option includes any standard abbreviations. Does not apply to U.S. addresses.
	U	Unmatched.
	V	Validated. The data was confirmed correct and remained unchanged from input.
City.Result	**A**	Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C	Corrected. U.S. and Canadian addresses only.
	F	Hyphens missing or punctuation errors. Canadian addresses only.
	M	Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. Does not apply to U.S. or Canadian addresses.
	P	Pass-through. The data was not used in the validation process, but it was preserved in the output.
	R	The city is required but is missing from the input address. U.S. addresses only.
	S	Standardized. This option includes any standard abbreviations. Does not apply to U.S. addresses.
	U	Unmatched. Does not apply to Canadian addresses.

Response Element	Description
	V Validated. The data was confirmed correct and remained unchanged from input.
Country.Result	These result codes do not apply to U.S. or Canadian addresses.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field.
	S Standardized. This option includes any standard abbreviations.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input.
FirmName.Result	**C** Corrected. U.S. addresses only.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. U.S. and Canadian addresses only.
	U Unmatched. U.S. and Canadian addresses only.
	V Validated. The data was confirmed correct and remained unchanged from input. U.S. addresses only.
HouseNumber.Result	**A** Appended. The field was added to a blank input field. Canadian addresses only.
	C Corrected. Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About Additional Input Data**.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	O Out of range. Does not apply to U.S. or Canadian addresses.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only.
	R The house number is required but is missing from the input address. Canadian addresses only.
	S Standardized. This option includes any standard abbreviations. Does not apply to U.S. or Canadian addresses.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input.
LeadingDirectional.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. Non-blank input was corrected to a non-blank value. U.S. addresses only.

Response Element	Description
	D Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About Additional Input Data**.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only.
	S Standardized. This option includes any standard abbreviations.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input. Does not apply to Canadian addresses.
POBox.Result	**A** Appended. The field was added to a blank input field. Canadian addresses only.
	C Corrected. Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About Additional Input Data**.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	M Multiple matches. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only.
	R The P.O. Box number is required but is missing from the input address. U.S. addresses only.
	S Standardized. This option includes any standard abbreviations.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input.
PostalCode.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. U.S. and Canadian addresses only.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.

Response Element	Description
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. Does not apply to Canadian addresses.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Does not apply to U.S. addresses.
	R The postal code is required but is missing from the input address. U.S. addresses only.
	S Standardized. This option includes any standard abbreviations. Does not apply to U.S. or Canadian addresses.
	U Unmatched. For example, if the street name does not match the postal code, both StreetName.Result and PostalCode.Result will contain U.
	V Validated. The data was confirmed correct and remained unchanged from input.
PostalCode.Type	**P** The ZIP Code™ contains only PO Box addresses. U.S. addresses only.
	U The ZIP Code™ is a unique ZIP Code™ assigned to a specific company or location. U.S. addresses only.
	M The ZIP Code™ is for military addresses. U.S. addresses only.
	null The ZIP Code™ is a standard ZIP Code™.
RRHC.Type	These result codes apply to U.S. addresses only.
	HC The address is a Highway Contract address.
	RR The address is a Rural Route address.
StateProvince.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. U.S. addresses only.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. Does not apply to U.S. or Canadian addresses.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. U.S. and Canadian addresses only.
	R The state is required but is missing from the input address. U.S. addresses only.
	S Standardized. This option includes any standard abbreviations. Does not apply to U.S. addresses.
	U Unmatched. Does not apply to Canadian addresses.
	V Validated. The data was confirmed correct and remained unchanged from input.

Response Element	Description
Street.Result	These result codes apply to international addresses only.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output.
	R Street corrected. House number is out of range. Applies to UK and Japanese records only.
	S Standardized. This option includes any standard abbreviations.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input.
StreetName.Result	**A** Appended. The field was added to a blank input field. Canadian addresses only.
	C Corrected. U.S. and Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. addresses only. For more information, see **About Additional Input Data**.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Does not apply to U.S. addresses.
	S Standardized. This option includes any standard abbreviations. U.S. and Canadian addresses only.
	U Unmatched.
	V Validated. The data was confirmed correct and remained unchanged from input.
StreetSuffix.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. U.S. and Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. and Canadian addresses only. For more information, see **About Additional Input Data**.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.

Response Element	Description
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only.
	S Standardized. This option includes any standard abbreviations.
	U Unmatched. Does not apply to U.S. addresses.
	V Validated. The data was confirmed correct and remained unchanged from input.
TrailingDirectional.Result	**A** Appended. The field was added to a blank input field. U.S. and Canadian addresses only.
	C Corrected. U.S. and Canadian addresses only.
	D Dropped. The field provided on input was removed. U.S. and Canadian addresses only. For more information, see **About Additional Input Data**.
	F Formatted. The spacing and/or punctuation was changed to conform to postal standards. Does not apply to U.S. or Canadian addresses.
	M Multiple. The input address matched multiple records in the postal database, and each matching record has a different value in this field. U.S. addresses only.
	P Pass-through. The data was not used in the validation process, but it was preserved in the output. Canadian addresses only.
	S Standardized. This option includes any standard abbreviations.
	U Unmatched. Does not apply to Canadian addresses.
	V Validated. The data was confirmed correct and remained unchanged from input.

Universal Name Module

OpenNameParser

OpenNameParser breaks down personal and business names and other terms in the name data field into their component parts. These parsed name elements are then subsequently available to other automated operations such as name matching, name standardization, or multi-record name consolidation.

OpenNameParser does the following:

- Determines the type of a name in order to describe the function that the name performs. Name entity types are divided into two major groups: personal names and business names. Within each of these major groups are subgroups.
- Determines the form of a name in order to understand which syntax the parser should follow for parsing. Personal names usually take on a natural (signature) order or a reverse order. Business names are usually ordered hierarchically.

- Determines and labels the component parts of a name so that the syntactical relationship of each name part to the entire name is identified. The personal name syntax includes prefixes, first, middle, and last name parts, suffixes, and account description terms, among other personal name parts. The business name syntax includes the firm name and suffix terms.
- Parses conjoined personal and business names and either retains them as one record or splits them into multiple records. Examples of conjoined names include "Mr. and Mrs. John Smith" and "Baltimore Gas & Electric dba Constellation Energy".
- Parses output as records or as a list.
- Assigns a parsing score that reflects the degree of confidence that the parsing is correct.

Resource URL

JSON endpoint:

```
http://server:port/rest/OpenNameParser/results.json
```

XML endpoint:

```
http://server:port/rest/OpenNameParser/results.xml
```

Example with JSON Response

The following example requests a JSON response:

```
http://myserver:8080/rest/OpenNameParser/results.json?
Data.Name=John+Williams+Smith
```

The JSON returned by this request would be:

```
{"output_port": [{
   "Name": "John Williams Smith",
   "CultureCodeUsedToParse": "",
   "FirstName": "John",
   "LastName": "Smith",
   "MiddleName": "Williams",
   "Names": [],
   "IsParsed": true,
   "IsPersonal": true,
   "IsConjoined": false,
   "IsReverseOrder": false,
   "IsFirm": false,
   "NameScore": 100,
   "user_fields": []
}]}
```

Example with XML Response

The following example requests an XML response:

```
http://myserver:8080/rest/OpenNameParser/results.xml?
Data.Name=John+Williams+Smith
```

The XML returned by this request would be:

```
<ns2:xml.OpenNameParserResponse
xmlns:ns2="http://www.pb.com/spectrum/services/OpenNameParser">
   <ns2:output_port>
      <ns2:Result>
         <ns2:Name>John Williams Smith</ns2:Name>
         <ns2:CultureCodeUsedToParse/>
         <ns2:FirstName>John</ns2:FirstName>
         <ns2:LastName>Smith</ns2:LastName>
         <ns2:MiddleName>Williams</ns2:MiddleName>
         <ns2:Names/>
         <ns2:IsParsed>true</ns2:IsParsed>
         <ns2:IsPersonal>true</ns2:IsPersonal>
         <ns2:IsConjoined>false</ns2:IsConjoined>
         <ns2:IsReverseOrder>false</ns2:IsReverseOrder>
         <ns2:IsFirm>false</ns2:IsFirm>
         <ns2:NameScore>100</ns2:NameScore>
         <ns2:user_fields/>
      </ns2:Result>
   </ns2:output_port>
</ns2:xml.OpenNameParserResponse>
```

Request

Parameters for Input Data

Table 117: Open Name Parser Input

Parameter	Description
Data.CultureCode	The culture of the input name data. The options are listed below. **Null (empty)** Global culture (default). **de** German. **es** Spanish. **ja** Japanese. **Note:** If you added your own domain using the Open Parser Domain Editor, the cultures and culture codes for that domain are also valid.
Data.Name	The name you want to parse. This field is required.

Options

Parameters for Parsing Options

The following table lists the options that control the parsing of names.

Table 118: Open Name Parser Parsing Options

Parameter	Description
Option.ParseNaturalOrderPersonalNames	Specifies whether to parse names where the is in the order Title, First Name, Middle Name, Last Name, and Suffix. **true** Parse personal names that are in natural order. **false** Do not parse names that are in natural order.
Option.ParseReverseOrderPersonalNames	Specifies whether to parse names where the last name is specified first. **true** Parse personal names that are in reverse order.

Parameter	Description
	false Do not parse names that are in reverse order.
Option.ParseConjoinedNames	Specifies whether to parse conjoined names. **true** Parse conjoined names. **false** Do not parse conjoined names.
Option.SplitConjoinedNames	Specifies whether to separate names containing more than one individual into multiple records, for example, `Bill & Sally Smith`. **true** Split conjoined names. **false** Do not split conjoined names.
Option.ParseBusinessNames	Specifies whether to parse business names. **true** Parse business names. **false** Do not parse business names.
Option.OutputAsList	Specifies whether to return the parsed name elements in a list form. **true** Return the parsed elements in a list form. **false** Do not return the parsed elements in a list form.
Option.ShortcutThreshold	Specifies how to balance performance versus quality. A faster performance will result in lower quality output; likewise, higher quality will result in slower performance. When this threshold is met, no other processing will be performed on the record. Specify a value from 0 to 100. The default is 100.

Parameters for Culture Options

The following table lists the options that control name cultures.

Table 119: Open Name Parser Cultures Options

Parameter	Description
Option.DefaultCulture	Specifies which culture(s) you want to include in the parsing grammar. Global Culture is the default selection. Specify cultures by specifying the two-character culture code in a comma-separated list in priority order. For example, to attempt to parse the name using the Spanish culture first then Japanese, you would specify:

Parameter	Description
	`es,ja,,`

Parameters for Advanced Options

The following table lists the advanced options for name parsing.

Table 120: Open Name Parser Advanced Options

Option	Description
Option.NaturalOrderPersonalNamesDomain	Specifies the domain to use when parsing natural order personal names. The valid values are the domain names defined in the Open Parser Domain Editor too in Enterprise Designer.
Option.NaturalOrderPersonalNamesPriority	Specify a number between 1 and 5 that indicates the priority of the natural order personal names domain relative to the other domains that you are using. This determines the order in which you want the parsers to run. Results will be returned for the first domain that scores higher than the number set in the shortcut threshold option. If no domain reaches that threshold, results for the domain with the highest score are returned. If multiple domains reach the threshold at the same time, priority goes to the domain that was run first (determined by the order set here) and its results will be returned.
Option.ReverseOrderPersonalNamesDomain	Specifies the domain to use when parsing reverse order personal names. The valid values are the domain names defined in the Open Parser Domain Editor too in Enterprise Designer.
Option.ReverseOrderPersonalNamesPriority	Specify a number between 1 and 5 that indicates the priority of the reverse order personal names domain relative to the other domains that you are using. This determines the order in which you want the parsers to run. Results will be returned for the first domain that scores higher than the number set in the shortcut threshold option. If no domain reaches that threshold, results for the domain with the highest score are returned. If multiple domains reach the threshold at the same time, priority goes to the domain that was run first (determined by the order set here) and its results will be returned.
Option.NaturalOrderConjoinedPersonalNamesDomain	Specifies the domain to use when parsing natural order conjoined personal names. The valid values are the domain names defined in the Open Parser Domain Editor too in Enterprise Designer.

Option	Description
Option.NaturalOrderConjoinedPersonalNamesPriority	Specify a number between 1 and 5 that indicates the priority of the natural order conjoined personal names domain relative to the other domains that you are using. This determines the order in which you want the parsers to run. Results will be returned for the first domain that scores higher than the number set in the shortcut threshold option. If no domain reaches that threshold, results for the domain with the highest score are returned. If multiple domains reach the threshold at the same time, priority goes to the domain that was run first (determined by the order set here) and its results will be returned.
Option.ReverseOrderConjoinedPersonalNamesDomain	Specifies the domain to use when parsing reverse order conjoined personal names. The valid values are the domain names defined in the Open Parser Domain Editor too in Enterprise Designer.
Option.ReverseOrderConjoinedPersonalNamesPriority	Specify a number between 1 and 5 that indicates the priority of the reverse order conjoined personal names domain relative to the other domains that you are using. This determines the order in which you want the parsers to run. Results will be returned for the first domain that scores higher than the number set in the shortcut threshold option. If no domain reaches that threshold, results for the domain with the highest score are returned. If multiple domains reach the threshold at the same time, priority goes to the domain that was run first (determined by the order set here) and its results will be returned.
Option.BusinessNamesDomain	Specifies the domain to use when parsing business names. The valid values are the domain names defined in the Open Parser Domain Editor too in Enterprise Designer.
Option.BusinessNamesPriority	Specify a number between 1 and 5 that indicates the priority of the business names domain relative to the other domains that you are using. This determines the order in which you want the parsers to run. Results will be returned for the first domain that scores higher than the number set in the shortcut threshold option. If no domain reaches that threshold, results for the domain with the highest score are returned. If multiple domains reach the threshold at the same time, priority goes to the domain that was run first (determined by the order set here) and its results will be returned.

Response

Table 121: Open Name Parser Output

Response Element	Format	Description
AccountDescription	String	An account description that is part of the name. For example, in "Mary Jones Account # 12345", the account description is "Account#12345".
Names	String	A hierarchical field that contains a list of parsed elements. This field is returned when you check the **Output results as list** box under Parsing Options.
Fields Related to Names of Companies		
FirmConjunction	String	Indicates that the name of a firm contains a conjunction such as "d/b/a" (doing business as), "o/a" (operating as), and "t/a" (trading as).
FirmName	String	The name of a company. For example, "Pitney Bowes".
FirmSuffix	String	The corporate suffix. For example, "Co." and "Inc."
IsFirm	String	Indicates that the name is a firm rather than an individual.
Fields Related to Names of Individual People		
Conjunction	String	Indicates that the name contains a conjunction such as "and", "or", or "&".
CultureCode	String	The culture codes contained in the input data.
CultureCodeUsedToParse	String	Identifies the culture-specific grammar that was used to parse the data. **Null (empty)** Global culture (default). **de** German. **es** Spanish. **ja** Japanese. **Note:** If you added your own domain using the Open Parser Domain Editor, the cultures and culture codes for that domain will appear in this field as well.
FirstName	String	The first name of a person.
GeneralSuffix	String	A person's general/professional suffix. For example, MD or PhD.
IsParsed	String	Indicates whether an output record was parsed. Values are true or false.
IsPersonal	String	Indicates whether the name is an individual rather than a firm. Values are true or false.
IsReverseOrder	String	Indicates whether the input name is in reverse order. Values are true or false.
LastName	String	The last name of a person. Includes the paternal last name.
LeadingData	String	Non-name information that appears before a name.

Response Element	Format	Description
MaturitySuffix	String	A person's maturity/generational suffix. For example, Jr. or Sr.
MiddleName	String	The middle name of a person.
Name.	String	The personal or firm name that was provided in the input.
NameScore	String	Indicates the average score of known and unknown tokens for each name. The value of NameScore will be between 0 and 100, as defined in the parsing grammar. 0 is returned when no matches are returned.
SecondaryLastName	String	In Spanish parsing grammar, the surname of a person's mother.
TitleOfRespect	String	Information that appears before a name, such as "Mr.", "Mrs.", or "Dr."
TrailingData	String	Non-name information that appears after a name.
Fields Related to Conjoined Names		
Conjunction2	String	Indicates that a second, conjoined name contains a conjunction such as "and", "or", or "&".
Conjunction3	String	Indicates that a third, conjoined name contains a conjunction such as "and", "or", or "&".
FirmName2	String	The name of a second, conjoined company. For example, Baltimore Gas & Electric dba Constellation Energy.
FirmSuffix2	String	The suffix of a second, conjoined company.
FirstName2	String	The first name of a second, conjoined name.
FirstName3	String	The first name of a third, conjoined name.
GeneralSuffix2	String	The general/professional suffix for a second, conjoined name. For example, MD or PhD.
GeneralSuffix3	String	The general/professional suffix for a third, conjoined name. For example, MD or PhD.
IsConjoined	String	Indicates that the input name is conjoined. An example of a conjoined name is "John and Jane Smith".
LastName2	String	The last name of a second, conjoined name.
LastName3	String	The last name of a third, conjoined name.
MaturitySuffix2	String	The maturity/generational suffix for a second, conjoined name. For example, Jr. or Sr.
MaturitySuffix3	String	The maturity/generational suffix for a third, conjoined name. For example, Jr. or Sr.
MiddleName2	String	The middle name of a second, conjoined name.
MiddleName3	String	The middle name of a third, conjoined name.
TitleOfRespect2	String	Information that appears before a second, conjoined name, such as "Mr.", "Mrs.", or "Dr."

Response Element	Format	Description
TitleOfRespect3	String	Information that appears before a third, conjoined name, such as "Mr.", "Mrs.", or "Dr."

Appendix

In this section:

A

The ACR Code

In this section:

The ACR Code

The Address Correction Result (ACR) code describes what data has been changed in each record. An example of an ACR is:

L5-P0S0A5T1R0Z0C4-098

ACR codes consist of three parts:

- Validation Level
- Component Status
- Validation Match Score

Validation Level

The first two characters of the address correction result state the type and level of validation.

The first character, which is always alphabetic, specifies the type of validation:

- **U**—Unable to standardize address
- **C**—Address is in component form
- **L**—Address has been formatted into address lines
- **R**—Address has been reverted and has not reached acceptable level

The second character, which is always numeric, specifies the level of validation. The higher the level, the better the validation will be. The levels that can be achieved are as follows:

- **0**—No components validated
- **1**—Country only validated
- **2**—City and country validated
- **3**—City, postal code and country validated
- **4**—Street, city, postal code and country validated
- **5**—Premise number, building name, sub-building, PO box, company, street, city, postal code, and country validated.

Component Status

The second part of the ACR code gives the status of the main address components. The address components are identified as follows:

- Character 3-4: P—Premise/house number
- Character 5-6: S—Street
- Character 7-8: A—Subcity (city area)
- Character 9-10: T—City
- Character 11-12: R—Region/state
- Character 13-14: Z—Postal code/ZIP Code®
- Character 15-16: C—Country

A number follows each component and can take one of the following values:

- **0**—Not found/empty
- **1**—Derived using position in input data
- **2**—Recognized using the Address Now Module database
- **3**—Recognized and updated to standard form using the Address Now Module database
- **4**—Validated using Address Now Module database
- **5**—Updated/corrected using Address Now Module database
- **6**—Added using Address Now Module database
- **7**—Correctly empty

- **8**—Partial recognition using Address Now Module database
- **9**—Needs correcting to match Address Now Module database

Validation Match Score

The Validation Match Score comprises characters 17-19, the final three digits of the ACR code. This is a comparison between the standardized data (in component format) and the suggested match returned from the Address Now Module database.

This score is calculated by examining all fields returned from the Address Now Module database and comparing them individually with the existing component data. The overall match score is then calculated by combining these individual values into an average score, taking into account the match score weightings, which can be set from the address validation options dialog box. For example,

Input data:

AddressLine1: 11 High Street
City: Anytown
Country: UK

Standardized data:

Premise: 11
Street: High Street
City: Anytown

When validated, the data returned from the Address Now Module database for this record may be:

Premise: 11
Street: High Street
City: Anytown
Postal Code: ZZ9 9ZZ

Comparing the Address Now Module database to the standardized data we get:

- Premise: 100% match
- Street: 100% match
- City: 100% match
- Postal Code: not used, because empty on input

Combining these percentages gives us a match score of 100%.

Another example may be:

Input data:

AddressLine1: bergerstrasse 12
AddressLine2: munich
AddressLine3: 80124
Country: Germany

Standardized data:

Premise: 12
Street: Bergerstr.
City: München
Postal Code: 80124

Address Now Module database output:

Premise: 12
Street: Burgerstr.
City: München
Postal Code: 80142

Comparing the Address Now Module database output to the standardized data we get:

- Premise:100% match
- Street: 90% match (the actual figure is determined by a textual comparison of the two values)
- City: 100% match
- Postal Code: 80% match (because the numbers are transposed)

This gives an overall match score of 92% if the match score weightings are all set at 1. Increasing the match score weighting of the postal code will decrease the match score, because the postal code component score (80%) will be made more important in the calculation. Increasing the match score weighting of the city will increase the match score, because the city component score (100%) will be made more important.

For example:

L5-P4S4A5T5R4Z4C4-098

- L shows that formatting has been carried out to create the address lines
- The validation level is 5, meaning that the highest level of matching against the Address Now Module database was attained
- All component codes except subcity (A) and city (T) are set to 4 indicating that they were validated using the Address Now Module database
- The subcity code and city code are set to 5 indicating that these components were corrected using the Address Now Module database

The overall address matched the Address Now Module database at 98%.

Note: You may also receive a value of "SDS" for the Validation Match Score. A return of SDS indicates that the address has not been standardized, possibly as a result of the address being reverted.

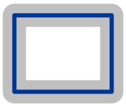
B

Buffering

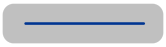
In this section:

Buffering

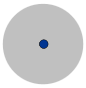
Use buffering to define areas that are close to the edges of a polygon, line, or point.



Buffered Polygon (zone)

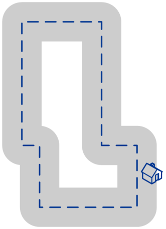


Buffered Line (corridor)

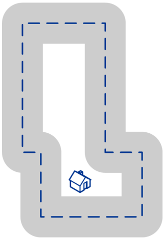


Buffered Point (circle)

For example, if you work for an insurance company you may want to know if a potential customer's house is within 500 feet of a flood plain so that you can suggest that they buy flood insurance even though they are not actually within the flood plain. The following illustration shows this scenario using a buffered polygon. The dotted line indicates the boundary of the flood plain and the shaded area shows a 500-foot buffer zone around the boundary.

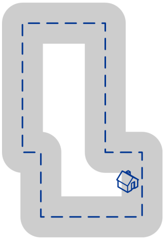


The buffer area extends on both sides of the boundary (inside and outside). When you use buffering, the output field BufferRelation indicates whether or not the point is in the buffered zone, and whether the point is inside or outside of the polygon, as shown in the following illustrations.



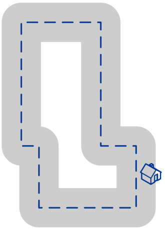
The point is inside the polygon and not in the buffer area.

The output field BufferRelation will contain "P".



The point is inside the polygon and in the buffer area.

The output field BufferRelation will contain "I".



The point is outside the polygon but in the buffer area.

The output field BufferRelation will contain "B".

Specify the size of polygon buffers using the BufferWidth input field to set it on a record-by-record basis and the option to set a default polygon buffer width for the job.

C

Country Codes

In this section:

Country ISO Codes and Module Support

The following table lists the ISO codes for each country as well as the modules that support addressing, geocoding, and routing for each country.

Note that the Enterprise Geocoding Module includes databases for Africa (30 countries), Middle East (8 countries) and Latin America (20 countries). These databases cover the smaller countries in those regions that do not have their own country-specific geocoding databases. The Supported Modules column indicates which countries are covered by these Africa, Middle East, and Latin America databases.

Also, the Geocode Address World database provides geographic and limited postal geocoding (but not street-level geocoding) for all countries.

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Afghanistan	AF	AFG	Address Now Module Universal Addressing Module
Aland Islands	AX	ALA	Address Now Module Universal Addressing Module
Albania	AL	ALB	Address Now Module Universal Addressing Module
Algeria	DZ	DZA	Address Now Module Universal Addressing Module
American Samoa	AS	ASM	Address Now Module Universal Addressing Module
Andorra	AD	AND	Address Now Module Enterprise Geocoding Module[1] Universal Addressing Module GeoComplete Module
Angola	AO	AGO	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Anguilla	AI	AIA	Address Now Module Universal Addressing Module
Antarctica	AQ	ATA	Address Now Module Universal Addressing Module
Antigua And Barbuda	AG	ATG	Address Now Module Universal Addressing Module
Argentina	AR	ARG	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Armenia	AM	ARM	Address Now Module Universal Addressing Module

[1] Andorra is covered by the Spain geocoder

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Aruba	AW	ABW	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Australia	AU	AUS	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Austria	AT	AUT	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Azerbaijan	AZ	AZE	Address Now Module Universal Addressing Module
Bahamas	BS	BHS	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Bahrain	BH	BHR	Address Now Module Enterprise Geocoding Module (Middle East) Enterprise Routing Module Universal Addressing Module
Bangladesh	BD	BGD	Address Now Module Universal Addressing Module
Barbados	BB	BRB	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Belarus	BY	BLR	Address Now Module Universal Addressing Module
Belgium	BE	BEL	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Belize	BZ	BLZ	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Benin	BJ	BEN	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Bermuda	BM	BMU	Address Now Module Universal Addressing Module
Bhutan	BT	BTN	Address Now Module Universal Addressing Module
Bolivia, Plurinational State Of	BO	BOL	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Bonaire, Saint Eustatius And Saba	BQ	BES	Address Now Module Universal Addressing Module
Bosnia And Herzegovina	BA	BIH	Address Now Module Universal Addressing Module
Botswana	BW	BWA	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Bouvet Island	BV	BVT	Address Now Module Universal Addressing Module
Brazil	BR	BRA	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
British Indian Ocean Territory	IO	IOT	Address Now Module Universal Addressing Module
Brunei Darussalam	BN	BRN	Address Now Module Universal Addressing Module
Bulgaria	BG	BGR	Address Now Module Universal Addressing Module
Burkina Faso	BF	BFA	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Burundi	BI	BDI	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Cambodia	KH	KHM	Address Now Module Universal Addressing Module
Cameroon	CM	CMR	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Canada	CA	CAN	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Cape Verde	CV	CPV	Address Now Module Universal Addressing Module
Cayman Islands	KY	CYM	Address Now Module Universal Addressing Module
Central African Republic	CF	CAF	Address Now Module Universal Addressing Module
Chad	TD	TCD	Address Now Module Universal Addressing Module
Chile	CL	CHL	Address Now Module Enterprise Geocoding Module Universal Addressing Module GeoComplete Module
China	CN	CHN	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Christmas Island	CX	CXR	Address Now Module Universal Addressing Module
Cocos (Keeling) Islands	CC	CCK	Address Now Module Universal Addressing Module
Colombia	CO	COL	Address Now Module Universal Addressing Module
Comoros	KM	COM	Address Now Module Universal Addressing Module
Congo	CG	COG	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Congo, The Democratic Republic Of The	CD	COD	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Cook Islands	CK	COK	Address Now Module Universal Addressing Module
Costa Rica	CR	CRI	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Côte d'Ivoire	CI	CIV	Address Now Module Universal Addressing Module
Croatia	HR	HRV	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Cuba	CU	CUB	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Curacao	CW	CUW	Address Now Module Universal Addressing Module
Cyprus	CY	CYP	Address Now Module Universal Addressing Module
Czech Republic	CZ	CZE	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Denmark	DK	DNK	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Djibouti	DJ	DJI	Address Now Module Universal Addressing Module
Dominica	DM	DMA	Address Now Module Universal Addressing Module
Dominican Republic	DO	DOM	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Ecuador	EC	ECU	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Egypt	EG	EGY	Address Now Module Enterprise Geocoding Module (Middle East) Enterprise Routing Module Universal Addressing Module
El Salvador	SV	SLV	Address Now Module Enterprise Geocoding Module (Latin

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
			America) Universal Addressing Module
Equatorial Guinea	GQ	GNQ	Address Now Module Universal Addressing Module
Eritrea	ER	ERI	Address Now Module Universal Addressing Module
Estonia	EE	EST	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Ethiopia	ET	ETH	Address Now Module Universal Addressing Module
Falkland Islands (Malvinas)	FK	FLK	Address Now Module Universal Addressing Module
Faroe Islands	FO	FRO	Address Now Module Universal Addressing Module
Fiji	FJ	FJI	Address Now Module Universal Addressing Module
Finland	FI	FIN	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
France	FR	FRA	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
French Guiana	GF	GUF	Address Now Module Enterprise Geocoding Module [2] Universal Addressing Module
French Polynesia	PF	PYF	Address Now Module Universal Addressing Module
French Southern Territories	TF	ATF	Address Now Module Universal Addressing Module
Gabon	GA	GAB	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module

[2] French Guiana is covered by the France geocoder

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Gambia	GM	GMB	Address Now Module Universal Addressing Module
Georgia	GE	GEO	Address Now Module Universal Addressing Module
Germany	DE	DEU	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Ghana	GH	GHA	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Gibraltar	GI	GIB	Address Now Module Enterprise Geocoding Module [3]Universal Addressing Module
Greece	GR	GRC	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Greenland	GL	GRL	Address Now Module Universal Addressing Module
Grenada	GD	GRD	Address Now Module Universal Addressing Module
Guadeloupe	GP	GLP	Address Now Module Enterprise Geocoding Module [4] Universal Addressing Module
Guam	GU	GUM	Address Now Module Universal Addressing Module
Guatemala	GT	GTM	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Guernsey	GG	GGY	Address Now Module Universal Addressing Module
Guinea	GN	GIN	Address Now Module Universal Addressing Module
Guinea-Bissau	GW	GNB	Address Now Module Universal Addressing Module

[3] Gibraltar is covered by the Spain geocoder

[4] Guadeloupe is covered by the France geocode

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Guyana	GY	GUY	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Haiti	HT	HTI	Address Now Module Universal Addressing Module
Heard Island and McDonald Islands	HM	HMD	Address Now Module Universal Addressing Module
Holy See (Vatican City State)	VA	VAT	Address Now Module Enterprise Geocoding Module [5] Universal Addressing Module
Honduras	HN	HND	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Hong Kong	HK	HKG	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Hungary	HU	HUN	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Iceland	IS	ISL	Address Now Module Universal Addressing Module
India	IN	IND	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Indonesia	ID	IDN	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Iran, Islamic Republic Of	IR	IRN	Address Now Module Universal Addressing Module
Iraq	IQ	IRQ	Address Now Module Universal Addressing Module
Ireland	IE	IRL	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module

[5] The Vatican is covered by the Italy geocoder

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Isle Of Man	IM	IMN	Address Now Module Universal Addressing Module
Israel	IL	ISR	Address Now Module Universal Addressing Module
Italy	IT	ITA	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Jamaica	JM	JAM	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Japan	JP	JPN	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Jersey	JE	JEY	Address Now Module Universal Addressing Module
Jordan	JO	JOR	Address Now Module Universal Addressing Module
Kazakhstan	KZ	KAZ	Address Now Module Universal Addressing Module
Kenya	KE	KEN	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Kiribati	KI	KIR	Address Now Module Universal Addressing Module
Korea, Democratic People's Republic Of	KP	PRK	Address Now Module Universal Addressing Module
Korea, Republic Of	KR	KOR	Address Now Module Universal Addressing Module
Kosovo	KS	KOS	Address Now Module Universal Addressing Module GeoComplete Module
Kuwait	KW	KWT	Address Now Module Enterprise Geocoding Module (Middle East) Enterprise Routing Module Universal Addressing Module

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Kyrgyzstan	KG	KGZ	Address Now Module Universal Addressing Module
Lao People's Democratic Republic	LA	LAO	Address Now Module Universal Addressing Module
Latvia	LV	LVA	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Lebanon	LB	LBN	Address Now Module Enterprise Geocoding Module (Middle East) Universal Addressing Module
Lesotho	LS	LSO	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Liberia	LR	LBR	Address Now Module Universal Addressing Module
Libyan Arab Jamahiriya	LY	LBY	Address Now Module Universal Addressing Module
Liechtenstein	LI	LIE	Address Now Module Enterprise Geocoding Module [6] Enterprise Routing Module Universal Addressing Module GeoComplete Module
Lithuania	LT	LTU	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Luxembourg	LU	LUX	Address Now Module Enterprise Geocoding Module [7] Enterprise Routing Module Universal Addressing Module GeoComplete Module
Macao	MO	MAC	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Macedonia, Former Yugoslav Republic Of	MK	MKD	Address Now Module Universal Addressing Module
Madagascar	MG	MDG	Address Now Module Universal Addressing Module

[6] Liechtenstein is covered by the Switzerland geocoder

[7] Luxembourg is covered by the Belgium geocoder

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Malawi	MW	MWI	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Malaysia	MY	MYS	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Maldives	MV	MDV	Address Now Module Universal Addressing Module
Mali	ML	MLI	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Malta	ML	MLT	Enterprise Routing Module Address Now Module Universal Addressing Module
Marshall Islands	MH	MHL	Address Now Module Universal Addressing Module
Martinique	MQ	MTQ	Address Now Module Enterprise Geocoding Module Guadeloupe is covered by the France geocode Universal Addressing Module
Mauritania	MR	MRT	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Mauritius	MU	MUS	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Mayotte	YT	MYT	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Mexico	MX	MEX	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Micronesia, Federated States Of	FM	FSM	Address Now Module Universal Addressing Module
Moldova, Republic Of	MD	MDA	Address Now Module Universal Addressing Module

[8] Martinique is covered by the France geocoder.

[9] Mayotte is covered by the France geocoder.

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Monaco	MC	MCO	Address Now Module Enterprise Geocoding Module [10]Universal Addressing Module
Mongolia	MN	MNG	Address Now Module Universal Addressing Module
Montenegro	ME	MNE	Address Now Module Universal Addressing Module
Montserrat	MS	MSR	Address Now Module Universal Addressing Module
Morocco	MA	MAR	Address Now Module Enterprise Geocoding Module (Africa) Enterprise Routing Module Universal Addressing Module
Mozambique	MZ	MOZ	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Myanmar	MM	MMR	Address Now Module Universal Addressing Module
Namibia	NA	NAM	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Nauru	NR	NRU	Address Now Module Universal Addressing Module
Nepal	NP	NPL	Address Now Module Universal Addressing Module
Netherlands	NL	NLD	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
New Caledonia	NC	NCL	Address Now Module Universal Addressing Module
New Zealand	NZ	NZL	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Nicaragua	NI	NIC	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module

[10] Monaco is covered by the France geocoder

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Niger	NE	NER	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Nigeria	NG	NGA	Address Now Module Enterprise Geocoding Module (Africa) Enterprise Routing Module Universal Addressing Module
Niue	NU	NIU	Address Now Module Universal Addressing Module
Norfolk Island	NF	NFK	Address Now Module Universal Addressing Module
Northern Mariana Islands	MP	MNP	Address Now Module Universal Addressing Module
Norway	NO	NOR	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Oman	OM	OMN	Address Now Module Enterprise Geocoding Module (Middle East) Enterprise Routing Module Universal Addressing Module
Pakistan	PK	PAK	Address Now Module Universal Addressing Module
Palau	PW	PLW	Address Now Module Universal Addressing Module
Palestinian Territory, Occupied	PS	PSE	Address Now Module Universal Addressing Module
Panama	PA	PAN	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Papua New Guinea	PG	PNG	Address Now Module Universal Addressing Module
Paraguay	PY	PRY	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Peru	PE	PER	Address Now Module Enterprise Geocoding Module (Latin

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
			America) Universal Addressing Module
Philippines	PH	PHL	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Pitcairn	PN	PCN	Address Now Module Universal Addressing Module
Poland	PL	POL	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Portugal	PT	PRT	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Puerto Rico	PR	PRI	Address Now Module Universal Addressing Module
Qatar	QA	QAT	Address Now Module Enterprise Geocoding Module (Middle East) Enterprise Routing Module Universal Addressing Module
Reunion	RE	REU	Address Now Module Enterprise Geocoding Module [11]Universal Addressing Module
Romania	RO	ROU	Address Now Module Universal Addressing Module Enterprise Routing Module
Russian Federation	RU	RUS	Address Now Module Enterprise Geocoding Module Universal Addressing Module GeoComplete Module
Rwanda	RW	RWA	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Saint Barthelemy	BL	BLM	Address Now Module Universal Addressing Module

[11] Reunion is covered by the France geocoder

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Saint Helena, Ascension & Tristan Da Cunha	SH	SHE	Address Now Module Universal Addressing Module
Saint Kitts and Nevis	KN	KNA	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Saint Lucia	LC	LCA	Address Now Module Universal Addressing Module
Saint Martin (French Part)	MF	MAF	Address Now Module Universal Addressing Module
Saint Pierre and Miquelon	PM	SPM	Address Now Module Universal Addressing Module
Saint Vincent And The Grenadines	VC	VCT	Address Now Module Universal Addressing Module
Samoa	WS	WSM	Address Now Module Universal Addressing Module
San Marino	SM	SMR	Address Now Module Enterprise Geocoding Module [12] Universal Addressing Module
Sao Tome And Principe	ST	STP	Address Now Module Universal Addressing Module
Saudi Arabia	SA	SAU	Address Now Module Enterprise Geocoding Module (Middle East) Universal Addressing Module
Senegal	SN	SEN	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Serbia	RS	SRB	Address Now Module Universal Addressing Module
Seychelles	SC	SYC	Address Now Module Universal Addressing Module
Sierra Leone	SL	SLE	Address Now Module Universal Addressing Module
Singapore	SG	SGP	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module

[12] San Marino is covered by the Italy geocoder

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Sint Maarten (Dutch Part)	SX	SXM	Universal Addressing Module
Slovakia	SK	SVK	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Slovenia	SI	SVN	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Solomon Islands	SB	SLB	Address Now Module Universal Addressing Module
Somalia	SO	SOM	Address Now Module Universal Addressing Module
South Africa	ZA	ZAF	Address Now Module Enterprise Geocoding Module Universal Addressing Module GeoComplete Module
South Georgia And The South Sandwich Islands	GS	SGS	Address Now Module Enterprise Geocoding Module Universal Addressing Module
South Sudan	SS	SSD	Address Now Module Universal Addressing Module
Spain	ES	ESP	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Sri Lanka	LK	LKA	Address Now Module Universal Addressing Module
Sudan	SD	SDN	Address Now Module Universal Addressing Module
Suriname	SR	SUR	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Svalbard And Jan Mayen	SJ	SJM	Address Now Module Universal Addressing Module
Swaziland	SZ	SWZ	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Sweden	SE	SWE	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Switzerland	CH	CHE	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
Syrian Arab Republic	SY	SYR	Address Now Module Universal Addressing Module
Taiwan, Province of China	TW	TWN	Enterprise Routing Module Address Now Module Universal Addressing Module
Tajikistan	TJ	TJK	Address Now Module Universal Addressing Module
Tanzania, United Republic Of	TZ	TZA	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Thailand	TH	THA	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
Timor-Leste	TL	TLS	Address Now Module Universal Addressing Module
Togo	TG	TGO	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Tokelau	TK	TKL	Address Now Module Universal Addressing Module
Tonga	TO	TON	Address Now Module Universal Addressing Module
Trinidad and Tobago	TT	TTO	Address Now Module Enterprise Geocoding Module (Latin America) Universal Addressing Module
Tunisia	TN	TUN	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Turkey	TR	TUR	Address Now Module Enterprise Geocoding Module

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
			Universal Addressing Module GeoComplete Module
Turkmenistan	TM	TKM	Address Now Module Universal Addressing Module
Turks And Caicos Islands	TC	TCA	Address Now Module Universal Addressing Module
Tuvalu	TV	TUV	Address Now Module Universal Addressing Module
Uganda	UG	UGA	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Ukraine	UA	UKR	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module
United Arab Emirates	AE	ARE	Address Now Module Enterprise Geocoding Module (Middle East) Enterprise Routing Module Universal Addressing Module
United Kingdom	GB	GBR	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
United States	US	USA	Address Now Module Enterprise Geocoding Module Enterprise Routing Module Universal Addressing Module GeoComplete Module
United States Minor Outlying Islands	UM	UMI	Address Now Module Universal Addressing Module
Uruguay	UY	URY	Address Now Module Enterprise Geocoding Module Universal Addressing Module
Uzbekistan	UZ	UZB	Address Now Module Universal Addressing Module
Vanuatu	VU	VUT	Address Now Module Universal Addressing Module
Venezuela, Bolivarian Republic Of	VE	VEN	Address Now Module Enterprise Geocoding Module Universal Addressing Module

ISO Country Name	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3	Supported Modules
Viet Nam	VN	VNM	Enterprise Routing Module Address Now Module Universal Addressing Module
Virgin Islands, British	VG	VGB	Address Now Module Universal Addressing Module
Virgin Islands, U.S.	VI	VIR	Address Now Module Universal Addressing Module
Wallis and Futuna	WF	WLF	Address Now Module Universal Addressing Module
Western Sahara	EH	ESH	Address Now Module Universal Addressing Module
Yemen	YE	YEM	Address Now Module Universal Addressing Module
Zambia	ZM	ZMB	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module
Zimbabwe	ZW	ZWE	Address Now Module Enterprise Geocoding Module (Africa) Universal Addressing Module

D

ValidateAddress Confidence Algorithm

In this section:

Introduction to the Validate Address Confidence Algorithm

ValidateAddress computes a confidence score for each validated address. This score describes how likely it is that the validated address is correct. Confidence code values range from 0 to 100, with a zero confidence level indicating no confidence and 100 indicating a very high level of confidence that the match results are correct. Confidence codes are calculated based on an algorithm that takes into account the match results for individual output fields. The output fields involved in this calculation include:

- Country
- City
- State
- PostalCode
- StreetName
- HouseNumber
- LeadingDirectional
- TrailingDirectional
- StreetSuffix
- ApartmentNumber

Each field has its own Weight in the algorithm. Additionally, for each field the match result could be labeled as Success, Failure, or Changed. ("Changed" refers to cases where the contents of the field have been corrected in order to get a match.) The match result—Success, Failure, or Changed—determines what the Factor is for that field. Thus, the calculation for the confidence code is a product of Weight by Factor as follows:

```
Confidence = (Weight * Factor) for City
  + (Weight * Factor) for Country
  + (Weight * Factor) for State
  + (Weight * Factor) for PostalCode
  + (Weight * Factor) for StreetName
  + (Weight * Factor) for HouseNumber
  + (Weight * Factor) for Directionals
  + (Weight * Factor) for Street Suffix
  + (Weight * Factor) for ApartmentNumber
```

Confidence Algorithm for U.S. and Canadian Addresses

The following table details the scoring and logic behind the ValidateAddress confidence algorithm for U.S. and Canadian addresses.

Table 122: Confidence Algorithm for U.S. and Canadian Addresses

Field	Weight/Match Score	Factor if Changed[13]	Factor If Filled[14]
Country	10	100%	0%
City	10	50%	75%
State	15	50%	75%
PostalCode	15	25%	25%
StreetName	15	50%	75%

Field	Weight/Match Score	Factor if Changed[13]	Factor If Filled[14]
HouseNumber	15	50%	75%
Directionals	10	50%	75%
StreetSuffix	5	50%	75%
ApartmentNumber	5	50%	75%

Confidence Algorithm for International Addresses

There are two confidence algorithms for addresses outside the U.S. and Canada, one for addresses in countries that use postal codes and one for addresses in countries that do not use postal codes.

The following table details the confidence algorithm for non-U.S. and non-Canadian addresses from countries that use postal codes.

Table 123: Confidence Algorithm for Countries With Postal Codes

Field	Weight/Match Score	Factor if Changed[15]	Factor If Filled[16]	Factor if Postal Data Unavailable
Country	11.1111111111111	100%	0%	0%
City	11.1111111111111	50%	75%[17]	0%
State	16.6666666666667	100%	100	80%
PostalCode	16.6666666666667	100%	100%	80%
StreetName	16.6666666666667	50%	75%	50%
HouseNumber	16.6666666666667	50%	75%	50%
Directionals	0	50%	75%	0%
StreetSuffix	5.55555555555556	50%	75%	50%
ApartmentNumber	5.55555555555556	50%	75%	50%

[17] If the country is a Category C country, this value is 50%. Countries fall into one of these categories:

- **Category A**—Enables the validation and correction of an address's postal code, city name, state/county name, street address elements, and country name.
- **Category B**—Enables the validation and correction of an address's postal code, city name, state/county name, and country name. It does not support the validation or correction of street address elements.
- **Category C**—Enables the validation and correction of the country name, and the validation of the format of the postal code.

The following table details confidence algorithm for countries that do not use postal codes.

Table 124: Confidence Algorithm for Countries Without Postal Codes

Field	Weight/Match Score	Factor if Changed[18]	Factor If Filled[19]	Factor if Postal Data Unavailable
Country	13.3333333333333	100%	0%	0%
City	13.3333333333333	50%	75%[20]	0%
State	20	100%	100	80%
StreetName	20	50%	75%	50%
HouseNumber	20	50%	75%	50%
Directionals	0	50%	75%	0%
StreetSuffix	6.66666666666667	50%	75%	50%
ApartmentNumber	6.66666666666667	50%	75%	50%

The following table lists countries without postal codes.

Table 125: Countries Without Postal Codes

Afghanistan	Albania	Angola
Anguilla	Bahamas	Barbados
Belize	Benin	Bhutan
Botswana	Burkina Faso	Burundi
Cameroon	Cayman Islands	Central African Rep.
Chad	Cocos Islands	Columbia
Comoros	Congo (Dem. Rep.)	Congo (Rep.)
Cote d'Ivoire	Korea (North)	Djibouti
Dominica	Equatorial Guinea	Eritrea
Fiji	Gabon	Gambia
Ghana	Grenada	Guyana

[20] If the country is a Category C country, this value is 50%. Countries fall into one of these categories:

- **Category A**—Enables the validation and correction of an address's postal code, city name, state/county name, street address elements, and country name.
- **Category B**—Enables the validation and correction of an address's postal code, city name, state/county name, and country name. It does not support the validation or correction of street address elements.
- **Category C**—Enables the validation and correction of the country name, and the validation of the format of the postal code.

Ireland	Jamaica	Kiribati
Libya	Malawi	Mali
Mauritania	Namibia	Nauru
Palaos	Panama	Peru
Qatar	Rwanda	Saint Lucia
Saint Vincent & Grenadines	Samoa	Sao Tome & Principe
Seychelles	Sierra Leone	Suriname
Tanzania	Timor	Togo
Tonga	Trinidad & Tobago	Tuvalu
Uganda	United Arab Emirates	Vanuatu
Yemen	Zimbabwe	

Notices

USPS® Notices

Data Provider and Related Notices

ICU Notices