

# Spectrum Technology Platform

バージョン 2019.1.0

Web サービス ガイド



# 目次

## 1 - はじめに

---

REST	4
SOAP	28

## 2 - Web サービス

---

CORS の有効化	48
REST	49
SOAP	418

## 第章 : 付録

---

付録 A :	
バッファリング	731
付録 B :	
国コード	734
付録 C :	
ValidateAddress 確信アルゴリズム	761

# 1 - はじめに

## このセクションの構成

---

REST	4
SOAP	28

# REST

## REST インターフェイス

Spectrum™ Technology Platform は、Web サービスに対する REST インターフェイスを提供します。Enterprise Designer で作成されたユーザ定義の Web サービスは、GET メソッドと POST メソッドをサポートします。モジュールの一部としてインストールされているデフォルトのサービスは、GET しかサポートしません。POST を使用してこれらのサービスのいずれかにアクセスしたい場合は、Enterprise Designer でユーザ定義サービスを作成する必要があります。

ご利用の Spectrum™ Technology Platform サーバーで使用できる REST Web サービスを確認するには、以下の URL にアクセスします。

```
http://server:port/rest
```

注：URL の長さの制限により、パラメーターは 2,048 文字以下に制限することをお勧めします。

### サービスのエンドポイント

XML 応答のエンドポイント:

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

JSON 応答のエンドポイント:

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

ユーザ定義 Web サービスのエンドポイントは、別の URL を使用するよう Enterprise Designer で変更できます。

注：Spectrum™ Technology Platform では、デフォルトで 8080 番ポートを HTTP 通信に使用します。管理者が別のポートを設定している場合もあります。

### WADL の URL

Spectrum™ Technology Platform Web サービスの WADL:

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

例:

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

## ユーザフィールド

Web サービスでは使用されない余分なフィールドでも、サービスを介して渡すことができます。こうしたフィールドは、変更されずに応答の `user_fields` セクションで返されます。GET 要求の場合、ユーザフィールドは他のフィールドと同じように、URL におけるパラメータとして引き渡されます。POST 要求の場合、ユーザフィールドは、XML または JSON リクエストにおける `user_fields` 要素の一部として引き渡されます。

注：ユーザフィールド名に、XML や JSON 要素名で無効な文字を含めることはできません。例えば、スペースは有効ではありません。

## GET を用いて XML レスポンスを求める、REST リクエストの例

以下に、GET メソッドを使用して XML レスポンスを要求する、`ValidateAddress` サービスに対する REST リクエストの記述例を示します。

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

この要求の例では、XML 応答が要求されたので、以下の応答が返されます。

```
<?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>
```

## GET を用いて JSON レスポンスを求める、REST リクエストの例

以下に、GET メソッドを使用して JSON レスポンスを要求する、ValidateAddress サービスに対する REST リクエストの記述例を示します。

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

この要求の例では、JSON 応答が要求されたので、以下の応答が返されます。

```
{
  "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"
    }
  }
}
```

## JSON POST リクエスト

ユーザ定義の Web サービスを、REST Web サービスとしてエクスポートし、JSON 入力を受け取る POST メソッドを持つように設定できます。JSON POST リクエストには Content-Type:application/json を指定し、以下の形式を使用します。

### フラット データ

POST を使用してフラット データを Web サービスに送信するためのリクエスト本体の形式は次のとおりです。

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

```

    }
  ]
}
}

```

説明:

### **InputStageName**

Enterprise Designer のキャンバス上に表示される入力ステージの名前。デフォルトのステージ名は Input です。

### **InputDataType**

レコードレベルのエンティティに与えられる名前。この値は、データフローの Input ステージにおいて、[入力フィールド] タブの [データ タイプ名] フィールドで指定されます。デフォルトのレコードレベル エンティティ名は Row です。

### **FieldName1 と FieldName2**

サービスの Input ステージで定義されている入力フィールドの名前。

### **FieldValue1 と FieldValue2**

対応するフィールドに入力され、Web サービスに送信する入力データ。

## リスト データ

リスト データは、親フィールドの下でグループ化されているフィールドの階層グループ構造で構成されます。

注：リスト データを入力として使用するには、GET リソースを持たない REST Web サービスとしてサービスがエクスポートされている必要があります。サービスに GET リソースがある場合は、サービスのエクスポート時に Enterprise Designer においてエラーが発生します。GET が階層フィールドをサポートしないためです。

POST を用いてリスト データを Web サービスに送信するための形式は次のとおりです。

```

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

```

説明:

**InputStageName**

Enterprise Designer のキャンバス上に表示される入力ステージの名前。デフォルトのステージ名は Input です。

**InputDataType**

レコードレベルのエンティティに与えられる名前。この値は、データフローの Input ステージにおいて、**[入力フィールド]** タブの **[データ タイプ名]** フィールドで指定されます。デフォルトのレコードレベル エンティティ名は Row です。

**ListField1**

サービスの Input ステージで定義されている階層フィールドの名前。

**SubfieldName1 と SubfieldName2**

リスト フィールドを構成する子フィールドの名前。

**SubfieldValue1 と SubfieldValue2**

Web サービスに送信する入力データ。

**ユーザ フィールド**

Web サービスでは使用されない余分なフィールドを、サービスを介して渡すことができます。こうしたフィールドは、変更されずに応答の `user_fields` セクションで返されます。リクエストで供給するユーザ フィールドは、サービス データフローの Input ステージで定義する必要はありません。

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

説明:

**InputStageName**

Enterprise Designer のキャンバス上に表示される入力ステージの名前。デフォルトのステージ名は Input です。

**InputDataType**



レコードレベルのエンティティに与えられる名前。この値は、データフローの Input ステージにおいて、**[入力フィールド]** タブの **[データ タイプ名]** フィールドで指定されます。デフォルトのレコードレベル エンティティ名は Row です。

### **FieldName1 と FieldName2**

パススルー フィールドの名前。

### **FieldValue1 と FieldValue2**

パススルー フィールドへの入力データ。

## オプション

リクエストでオプションを指定することによって、サービス データフローで指定されたデフォルトのオプションをオーバーライドできます。ユーザ定義の Web サービスに対してオプションを指定できるのは、データフローがオプションを受け付けるように設定されている場合のみです。リクエストにおいてオプションを受け付けるようにサービスを設定するには、Enterprise Designer でそのサービスを開き、**[編集]** > **[データフロー オプション]** を選択します。

リクエストで処理オプションを指定するための形式は次のとおりです。

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

説明:

### **OptionName1**

オプション名。サービスに対する有効なオプションの一覧については、そのサービスの WADL を参照するか、Enterprise Designer でそのサービスを開いて **[編集]** > **[データフロー オプション]** を選択します。

### **OptionValue1**

オプションに対する有効な値。有効値の一覧については、Enterprise Designer でそのサービスを開いて **[編集]** > **[データフロー オプション]** を選択します。

### **POST を用いた JSON リクエストの例**

以下に、Web サービスに対する POST を用いた JSON リクエストにおいて、オプション、フラット フィールド、リスト フィールド、ユーザ定義フィールドを使用する例を示します。

```
{
  "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"
  }
]]
}
]
}
```

この例の内容は次のとおりです。

- OutputCasing は、Web サービスがエクスポートするオプションで、出力を大文字と小文字のどちらで返すかを制御します。この例では、大文字を表す U が設定されています。
- Input は、データフローの Input ステージのラベルです。Enterprise Designer のキャンバス上に表示されているものと同じです。
- Address は、レコードレベル エンティティの名前です。データフローの Input ステージにおいて、[入力フィールド] タブの [データ タイプ名] フィールドで指定されたものです。
- AddressLine1、City、および StateProvince は、フラット フィールドです。
- Accounts は、階層 ("リスト") フィールドです。AccountNumber と ExpirationDate というサブフィールド名が含まれています。この例では、2 つのアカウントが含まれています。
- user\_fields は、ユーザ定義フィールドを含みます。これらはパススルーされ、Web サービスによって変更されることなく出力に返されます。

## XML POST リクエスト

ユーザ定義の Web サービスを、REST Web サービスとしてエクスポートし、XML 入力を受け取る POST メソッドを持つように設定できます。XML POST リクエストには Content-Type:application/xml を指定し、以下の形式を使用します。

### フラット データ

POST を用いてフラット データを Web サービスに送信するための形式は次のとおりです。

```
<ServiceNameRequest
xmlns:svc="http://www.pb.com/spectrum/services/ServiceName">
  <svc:Input>
    <svc:Row>
      <svc:Field1>Example value</svc:Field1>
      <svc:Field2>Another example value</svc:Field2>
    </svc:Row>
  </svc:Input>
</ServiceNameRequest>
```

説明:

#### **serviceName,**

Spectrum™ Technology Platform サーバー上の Web サービスの名前。

#### **Field1 と Field2**

サービスの Input ステージで定義されている入力フィールドの名前。

たとえば、このリクエストは、名と姓を **CasingExample** という名前のサービスに送信します。

```
<CasingExampleRequest
xmlns:svc="http://www.pb.com/spectrum/services/CasingExample">
  <svc:Input>
    <svc:Row>
      <svc:FirstName>Alex</svc:FirstName>
      <svc:LastName>Smith</svc:LastName>
    </svc:Row>
  </svc:Input>
</CasingExampleRequest>
```

### リスト データ

リスト データは、親フィールドの下でグループ化されているフィールドの階層グループ構造で構成されます。

**注:** リスト データを入力として使用するには、GET リソースを持たない REST Web サービスとしてサービスがエクスポートされている必要があります。サービスに GET リソースがある場合は、サービスのエクスポート時に Enterprise Designer においてエラーが発生します。GET が階層フィールドをサポートしないためです。

POST を用いてリスト データを Web サービスに送信するための形式は次のとおりです。

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceNameRequest
  xmlns:svc="http://www.pb.com/spectrum/services/ServiceName">
  <svc:Input>
    <svc:Row>
      <svc>ListField1>
        <svc:DataType>
          <svc:SubField1>Example value</svc:SubField1>
          <svc:SubField2>Example value</svc:SubField2>
        </svc:DataType>
      </svc>ListField1>
    </svc:Row>
  </svc:Input>
</ServiceNameRequest>
```

説明:

### **ListField1**

サービスの Input ステージで定義されている階層フィールドの名前。

### **DataType**

サービスの Input ステージで定義されているリスト フィールドのデータ タイプ。

### **Subfield1 と Subfield2**

リスト フィールドを構成する子フィールドの名前。

たとえば、このリクエストは、名、姓、電話番号のリストを **CasingExample** という名前のサービスに送信します。

```
<CasingExampleRequest
  xmlns:svc="http://www.pb.com/spectrum/services/CasingExample">
  <svc:Input>
    <svc:Row>
      <svc:FirstName>George</svc:FirstName>
      <svc:LastName>Washington</svc:LastName>
      <svc:PhoneNumbers>
        <svc:PhoneNumbers>
          <svc:HomePhone>123-234-9876</svc:HomePhone>
          <svc:CellPhone>123-678-9012</svc:CellPhone>
          <svc:OfficePhone>123-987-6543</svc:OfficePhone>
        </svc:PhoneNumbers>
      </svc:PhoneNumbers>
    </svc:Row>
  </svc:Input>
</CasingExampleRequest>
```

## ユーザ フィールド

Web サービスでは使用されない余分なフィールドを、サービスを介して渡すことができます。こうしたフィールドは、変更されずに応答の `user_fields` セクションで返されます。リクエストで供給するユーザ フィールドは、サービス データフローの `Input` ステージで定義する必要はありません。

```
<ServiceNameRequest
xmlns:svc="http://www.pb.com/spectrum/services/ServiceName">
  <svc:Input>
    <svc:Row>
      <svc:user_fields>
        <svc:user_field>
          <svc:name>FieldName</svc:name>
          <svc:value>FieldValue</svc:value>
        </svc:user_field>
      </svc:user_fields>
    </svc:Row>
  </svc:Input>
</ServiceNameRequest>
```

説明:

### **FieldName**

パススルー フィールドの名前。

### **FieldValue**

パススルー フィールドに含まれる値。

たとえば、このリクエストは配偶者の名前をパススルー フィールドとして送信します。ユーザ フィールド名は `Spouse` で、フィールドの値は `Martha` です。

```
<CasingExampleRequest
xmlns:svc="http://www.pb.com/spectrum/services/CasingExample">
  <svc:Input>
    <svc:Row>
      <svc:FirstName>George</svc:FirstName>
      <svc:LastName>Washington</svc:LastName>
      <svc:PhoneNumbers>
        <svc:PhoneNumbers>
          <svc:HomePhone>123-123-1234</svc:HomePhone>
          <svc:CellPhone>123-456-4567</svc:CellPhone>
          <svc:OfficePhone>123-678-6789</svc:OfficePhone>
        </svc:PhoneNumbers>
      </svc:PhoneNumbers>
      <svc:user_fields>
        <svc:user_field>
          <svc:name>Spouse</svc:name>
          <svc:value>Martha</svc:value>
        </svc:user_field>
      </svc:user_fields>
    </svc:Row>
  </svc:Input>
</CasingExampleRequest>
```

```

    </svc:Row>
  </svc:Input>
</CasingExampleRequest>

```

## オプション

リクエストでオプションを指定することによって、サービス データフローで指定されたデフォルトのオプションをオーバーライドできます。ユーザ定義の Web サービスに対してオプションを指定できるのは、データフローがオプションを受け付けるように設定されている場合のみです。リクエストにおいてオプションを受け付けるようにサービスを設定するには、Enterprise Designer でそのサービスを開き、**[編集] > [データフロー オプション]** を選択します。

リクエストで処理オプションを指定するための形式は次のとおりです。

```

<ServiceNameRequest
  xmlns:svc="http://www.pb.com/spectrum/services/ServiceName">
  <svc:options>
    <svc:OptionName>OptionValue</svc:OptionName>
  </svc:options>
  <svc:Input>
    <svc:Row> ... </svc:Row>
  </svc:Input>
</ServiceNameRequest>

```

説明:

### OptionName

オプション名。サービスに対する有効なオプションの一覧については、そのサービスの WADL を参照するか、Enterprise Designer でそのサービスを開いて **[編集] > [データフロー オプション]** を選択します。

### OptionValue

オプションに対する有効な値。有効値の一覧については、Enterprise Designer でそのサービスを開いて **[編集] > [データフロー オプション]** を選択します。

たとえば、このリクエストはオプション OutputCasing を U に設定します。

```

<AddressValidationRequest
  xmlns:svc="http://www.pb.com/spectrum/services/AddressValidation">
  <svc:options>
    <svc:OutputCasing>U</svc:OutputCasing>
  </svc:options>
  <svc:Input>
    <svc:Row>
      <svc:FirstName>George</svc:FirstName>
      <svc:LastName>Washington</svc:LastName>
      <svc:AddressLine1>123 Main St.</svc:AddressLine1>
      <svc:City>Springfield</svc:City>
      <svc:StateProvince>MO</svc:City>
    </svc:Row>
  </svc:Input>
</AddressValidationRequest>

```

```
</svc:Input>
</AddressValidationRequest>
```

### POST を用いた XML リクエストの例

以下に、Web サービスに対する POST を用いた XML リクエストにおいて、オプション、フラット フィールド、リスト フィールド、ユーザ定義フィールドを使用する例を示します。

```
<CasingExampleRequest
xmlns:svc="http://www.pb.com/spectrum/services/CasingExample">

  <svc:options>
    <svc:OutputCasing>U</svc:OutputCasing>
  </svc:options>
  <svc:Input>
    <svc:Row>
      <svc:FirstName>George</svc:FirstName>
      <svc:LastName>Washington</svc:LastName>
      <svc:AddressLine1>1073 Maple</svc:AddressLine1>
      <svc:City>Batavia</svc:City>
      <svc:StateProvince>IL</svc:StateProvince>
      <svc:PhoneNumbers>
        <svc:PhoneNumbers>

<svc:HomePhone>123-123-1234</svc:HomePhone>

<svc:CellPhone>123-345-3456</svc:CellPhone>

<svc:OfficePhone>123-456-4567</svc:OfficePhone>
        </svc:PhoneNumbers>
      </svc:PhoneNumbers>
      <svc:user_fields>
        <svc:user_field>
          <svc:name>Spouse</svc:name>
          <svc:value>Martha</svc:value>
        </svc:user_field>
      </svc:user_fields>
    </svc:Row>
  </svc:Input>
</CasingExampleRequest>
```

この例の内容は次のとおりです。

- OutputCasing は、Web サービスがエクスポートするオプションで、出力を大文字と小文字のどちらで返すかを制御します。この例では、大文字を表す U が設定されています。
- Input は、データフローの Input ステージのラベルです。Enterprise Designer のキャンバス上に表示されているものと同じです。

- Row は、レコードレベル エンティティの名前です。データフローの Input ステージにおいて、[入力フィールド] タブの [データ タイプ名] フィールドで指定されたものです。
- FirstName、LastName、AddressLine1、City、および StateProvince はフラット フィールドです。
- PhoneNumbers は、階層 ("リスト") フィールドです。HomePhone、CellPhone、および OfficePhone というサブフィールド名が含まれています。
- user\_fields は、ユーザ定義フィールドを含みます。これらはパススルーされ、Web サービスによって変更されることなく出力に返されます。

## Web サービスの認証

Spectrum™ Technology Platform Web サービスは要求元に対し、有効なユーザ資格情報による認証を求めます。認証方法には、ベーシック認証とトークンベースの認証の 2 種類があります。

### ベーシック認証

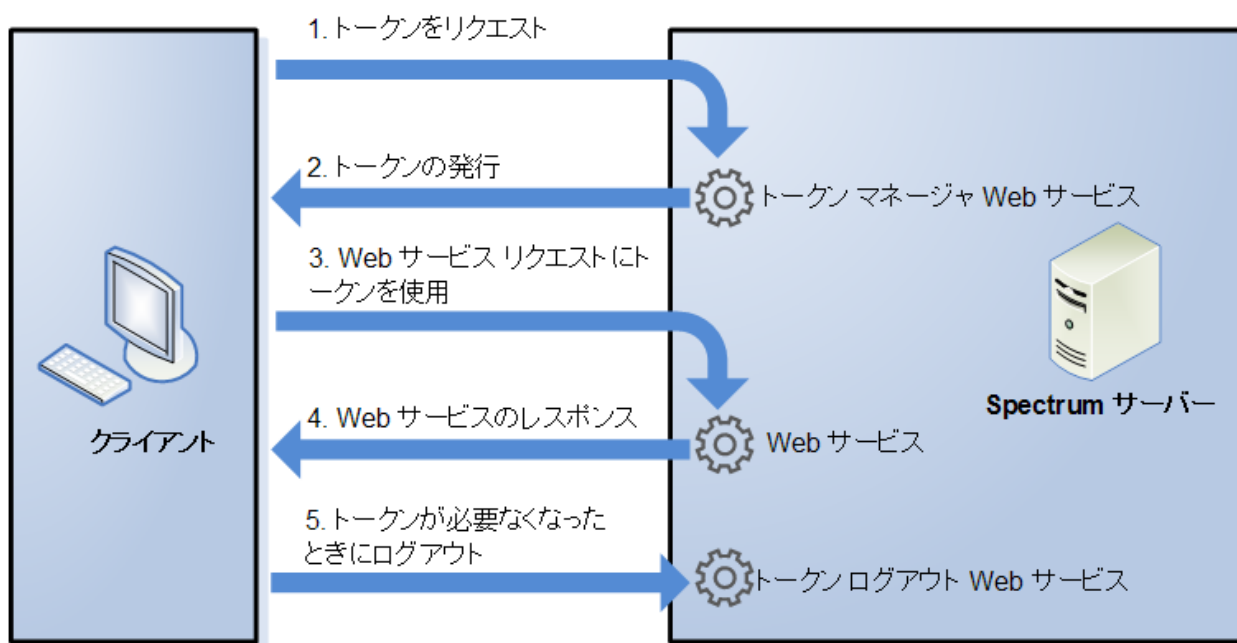
ベーシック認証では、ユーザ ID とパスワードが Web サービスへの各要求の HTTP ヘッダに付加されて Spectrum™ Technology Platform に引き渡されます。ベーシック認証はデフォルトで許可されていますが、管理者によってベーシック認証が無効化されている場合もあります。ベーシック認証が無効化されている場合は、トークンベースの認証を使用して Web サービスにアクセスする必要があります。

### トークンベースの認証

トークンベースの認証では、要求元は Spectrum™ Technology Platform サーバーからトークンを取得し、そのトークンを使用して Web サービスに要求を送信します。要求ごとにユーザの資格情報を送信する代わりに、トークンがサーバーに送信され、サーバーはトークンが有効であるかどうかを確認します。

次の図は、このプロセスを表したものです。





1. トークン管理サービスに要求を送信することにより、Spectrum™ Technology Platform サーバーからトークンを取得します。
2. トークン管理サービスがトークンを発行します。セッショントークンを要求した場合は、セッション ID も発行されます。
3. HTTP ヘッダにトークンを入れて、対象の Web サービスに要求を送信します。セッショントークンの場合は、セッション ID を HTTP ヘッダに含めます。
4. Web サービスが応答を発行します。そのトークンを使用して、同じ Web サービス、または Spectrum™ Technology Platform サーバー上の他の任意の Web サービスに対して、追加の Web サービス要求を送信することができます。1つのトークンで送信可能な Web サービス要求数に上限はありませんが、トークンに有効期限 (Time To Live と呼ばれます) がある場合は、Time To Live の時間が経過するとトークンは無効になります。セッショントークンの場合は、操作がない状態が 30 分間続くと無効になります。
5. トークンが不要になった場合は、トークンログアウト Web サービスに要求を送信することによって、ログアウトする必要があります。それによってトークンは、Spectrum™ Technology Platform サーバー上の有効トークンのリストから削除されます。

## トークン認証の使用

### トークンの取得

トークンを取得するには、security サーバー上の Spectrum™ Technology Platform Web サービスに要求を送信します。security WADL には、次の URL でアクセスできます。

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

この Web サービスはベーシック認証を使用するため、有効な Spectrum™ Technology Platform ユーザ名とパスワードを要求に含める必要があります。

security Web サービスは、2 種類のトークンを発行できます。トークンの種類は次のとおりです。

- セッション トークン
- オープン トークン

### セッション トークンの取得

セッショントークンは、ユーザセッションに関連付けられており、トークンを要求したコンピュータしか使用できません。セッションに関連付けられているため、セッションで操作のない状態が 30 分間続くと、このトークンは無効になります。セッション トークンは最もセキュリティが高く、Spectrum™ Technology Platform の認証に使用するトークンとして推奨されます。

セッション トークンを取得するには、次の URL を使用します。

```
http://server:port/security/rest/token/access/session/ttlInMinutes
```

説明:

**サーバー**

Spectrum™ Technology Platform サーバーのホスト名または IP アドレス。

**port**

Spectrum™ Technology Platform によって使用される HTTP ポート。デフォルトでは 8080 です。

**ttlInMinutes**

トークンが失効するまでの時間 (分単位)。トークンの Time To Live とも呼ばれます。トークンが失効しないようにするには、0 を指定します。

応答のサンプルを以下に示します。

```
{
  "access_token":
  "eyJlbnMiOiJBMlI4Q0JDLUhmTmJlU2IiwiaWYwXnIjoizGlyIn0..ESnq4JNEBbVMKycd139z0w.NFXAskvY0seX",
  "session": "09aa1fbb-71j3-43c7-ab8c-d800214283d4",
  "username": "admin"
}
```

応答には、次の要素が含まれます。

**access\_token** セキュリティ トークン。

**session** トークンが関連付けられているセッションのセッション ID。要求にセッション ID が含まれている場合のみ、トークンは受け付けられます。

**username** トークンの取得に使用する Spectrum™ Technology Platform ユーザ名。ユーザ名は、情報として返されるだけで、トークンを使用する時には必要ありません。

### オープントークンの取得

オープントークンは、ユーザにも特定のコンピュータにも関連付けられていません。これは、最もセキュリティの低いトークンです。

**重要:** 有効期限のないオープントークンは使用しないでください。オープントークンが承認されていない第三者によって取得されると、Spectrum™ Technology Platform サーバーへのアクセスに任意のコンピュータから無期限に使用される恐れがあります。

オープントークンを取得するには、次の URL を使用します。

```
http://server:port/security/rest/token/access/ttlInMinutes
```

説明:

**サーバー**

Spectrum™ Technology Platform サーバーのホスト名または IP アドレス。

**port**

Spectrum™ Technology Platform によって使用される HTTP ポート。デフォルトでは 8080 です。

**ttlInMinutes**

トークンが失効するまでの時間 (分単位)。トークンの Time To Live とも呼ばれます。トークンが失効しないようにするには、0 を指定します。

応答のサンプルを以下に示します。

```
{
  "access_token": "eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaWF0IjoiZGlyIn0..fI",
  "username": "admin"
}
```

応答には、次の要素が含まれます。

**access\_token** セキュリティ トークン。

**username** トークンの取得に使用する Spectrum™ Technology Platform ユーザ名。ユーザ名は、情報として返されるだけで、トークンを使用する時には必要ありません。

## トークンの使用

トークンを取得すると、要求にトークンを含めることによって、それを Spectrum™ Technology Platform Web サービスに対する認証に使用できます。Authorization HTTP ヘッダとして、または Cookie HTTP ヘッダとしての 2 つの方法でこれを実行できます。

**注：**1 つのトークンで実行可能な Web サービス要求数に上限はありませんが、有効期限があるトークンを要求した場合、トークンはいずれ失効します。セッショントークンの場合は、操作がない状態が 30 分間続くと無効になります。

## Authorization ヘッダにおけるトークンの使用

トークンを HTTP Authorization ヘッダで使用するには、次のフォーマットを使用します。

```
Authorization: Bearer Token
```

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Authorization: Bearer
eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
```

セッショントークンの場合は、セッション識別子も、Cookie ヘッダに次の形式で指定する必要があります。

```
Cookie: SESSION=SessionID
```

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Authorization: Bearer
eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
Cookie: SESSION=fff96e54-1615-4192-96c1-ea2f133ec6eb
```

**注：**クッキー名 SESSION はすべて大文字で記述する必要があります。

### Cookie ヘッダにおけるトークンの使用

Authorization ヘッダよりもクッキーを使用する方が容易であれば、トークンを Cookie ヘッダに次の形式で指定できます。

```
Cookie: spectrum.authentication.token=Token
```

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Cookie:
spectrum.authentication.token=eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
```

セッション トークンの場合は、セッション識別子も、Cookie ヘッダに次の形式で指定する必要があります。

```
Cookie: SESSION=SessionID
```

**注：**クッキー名 **SESSION** はすべて大文字で記述する必要があります。

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Cookie:
spectrum.authentication.token=eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
Cookie: SESSION=fff96e54-1615-4192-96c1-ea2f133ec6eb
```

### ログオフ

トークンの使用を終えたら、security Web サービスに要求を送信して、そのトークンを Spectrum™ Technology Platform サーバー上で管理されている有効なトークンのリストから削除する必要があります。security WADL には、次の URL でアクセスできます。

```
http://server:port/security/rest?wadl
```

ログアウトするには、次の URL を使用します。

```
http://server:port/security/rest/token/logout
```

トークンを、Authorization HTTP ヘッダまたは Cookie HTTP ヘッダに含めます。セッショントークンの場合は、セッションを Cookie ヘッダに含めます。詳細については、「[トークンの使用 \(40ページ\)](#)」を参照してください。このサービスにパラメータはありません。

## Web サービスとしてのサービスのエクスポート

Spectrum™ Technology Platform サービスは、RESTful および SOAP Web サービスとして使用できます。サービスをサーバー上で Web サービスとして使用できるようにするには

1. Enterprise Designer を開きます。
2. Web サービスとしてエクスポートするサービスを開きます。
3. **[編集]** > **[Web サービス オプション]** を選択します。
4. サービスを SOAP Web サービスとして使用できるようにするには、**[SOAP Web サービスとして公開]** チェック ボックスをオンにします。
5. サービスを REST Web サービスとして使用できるようにするには、**[REST Web サービスとして公開]** チェック ボックスをオンにして、次の手順を実行します。
  - a) デフォルトのエンドポイントをオーバーライドする場合は、使用するエンドポイントを **[パス]** フィールドに指定します。

パスの指定はオプションです。REST Web サービスのデフォルトのエンドポイントは次のとおりです。

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

別のエンドポイントを使用する場合は、指定したパスがサービス名の後に追加されます。例えば、"Americas/Shipping" と **[パス]** フィールドに指定すると、JSON エンドポイントは次のようになります。

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

**[変数の挿入]** ドロップダウンメニューをクリックして、使用するフィールドまたはオプションを選択することにより、フローからのフィールドとオプションをパス内の変数名として使用できます。変数はパス内において、`${Option.Name}` (フロー オプションの場合)、または `${Data.Name}` (フロー フィールドの場合) という表記で記述されます。

- b) REST Web サービスはデフォルトで、GET メソッドをサポートし、XML および JSON 形式でデータを返します。**[追加]** をクリックしてリソースを Web サービスに追加することによって、その他の HTTP メソッドや出力形式を定義できます。

リソースを追加する際に、HTTP メソッド (**GET** または **POST**) を選択できます。サポートされるデータ形式は以下のとおりです。これらの形式のすべてが使用できるとは限りま

せん。一部の形式は、お使いの Spectrum™ Technology Platform サーバー上に特定のモジュールがインストールされている場合のみ使用可能であるためです。

**XML** デフォルトの XML 形式。XML をリクエストとレスポンスの形式として使用し、処理するデータに特殊な XML 形式が存在しない場合は、この形式を使用します。

**JSON** デフォルトの JSON 形式。JSON をリクエストとレスポンスの形式として使用し、処理するデータに特殊な JSON 形式が存在しない場合は、この形式を使用します。

**GeoJSON** 地理的データを処理するサービスに適合した特殊な JSON 形式です。ジオメトリおよび次のネイティブプラットフォーム型でのみサポートされています。

- boolean
- double
- float
- integer
- bigdecimal
- long
- string
- date
- time
- datetime
- timespan

その他の型を持つフローをエクスポートしようとする場合は、GeoJSONを指定できません(設計時にエラーが表示されます)。また、GeoJSONでは単一ジオメトリのみを使用できます。出力に複数のジオメトリ フィールドが含まれている場合は、システムは "geometry" というフィールドに続いて "obj" というフィールドを検索します。そうしたフィールドが存在しない場合は、最初のジオメトリ フィールドが選択されます。

c) **[OK]** をクリックします。

新しいリソースが Web サービスに追加されます。

6. Web サービス オプションの設定を終えたら、**[OK]** をクリックします。

7. ツール バーにあるグレーの電球をクリックして、サービスをエクスポートします。

フローがエクスポートされると、次のように、Enterprise Designer ツール バーの電球ボタンがフローのエクスポートを示します。





サービスが Web サービスとしてエクスポートされているか確認するには、以下のいずれかの URL に移動します。

- REST の場合: `http://server:port/rest`
- SOAP の場合: `http://server:port/soap`

ここで、**server** は Spectrum™ Technology Platform サーバーの名前または IP アドレス、**port** は HTTP 通信に使用するポート番号です。

## REST Web サービスに対する POST サポートの追加

一部の Spectrum™ Technology Platform モジュールには、標準の Web サービスが付属しています。例えば、Universal Addressing モジュールには ValidateAddress Web サービスが付属しています。これらの Web サービスは GET のみをサポートします。このような Web サービスに POST のサポートを追加できます。Enterprise Designer でユーザー定義のサービスを作成し、その中に標準サービスをステージとして配置します。POST をサポートするユーザー定義サービスはエクスポート可能であるため、実質的に POST をサポートする標準サービスをエクスポートする Web サービスを作成していることになります。

1. Enterprise Designer を開きます。
2. [ファイル] > [新規作成] > [データフロー] > [サービス] に移動します。
3. **Input** ステージと **Output** ステージをキャンバス上にドラッグします。
4. サービスをキャンバス上にドラッグし、**Input** ステージと **Output** ステージをそれに接続します。

例えば、POST をサポートする ValidateAddress をエクスポートする場合、データフローは次のようになります。



5. 各ステージを設定します。
6. [編集] > [Web サービス オプション] を選択します。
7. [REST Web サービスとして公開] をオンにします。
8. [追加] をクリックして [POST] を選択し、入力と出力のフォーマットを選択します。
9. [OK] をクリックし、もう一度 [OK] をクリックします。

注: Web サービスのオプションを設定する詳しい手順については、[Web サービスとしてのサービスのエクスポート](#) (42ページ) を参照してください。



10. サービスを保存してエクスポートします。

以上で、POST をサポートする REST Web サービスとして標準サービスをエクスポートする、ユーザ定義の Web サービスが作成されます。

## マイクロバッチ処理

マイクロバッチ処理は、サーバーに対する1回のリクエストに複数のレコードを含める手法です。各レコードを個別にリクエストする代わりに、リクエストの中に複数のレコードを含めることにより、サービスによってレコードの大規模コレクションを処理する場合のパフォーマンスを大幅に向上させることができます。Spectrum™ Technology Platform では、REST および SOAP Web サービスと、Client SDK に対するマイクロバッチ処理がサポートされています。

### マイクロバッチ サイズ

1 回のリクエストに含めることのできるレコード数に制限はありませんが、一般的に 1 回のマイクロバッチで送信するレコード数を 50 ~ 100 にすると最良のパフォーマンスが得られます。さまざまなサイズのマイクロバッチをテストして、お使いの環境における最適なマイクロバッチ サイズを確認することをお勧めします。各入力レコードのレスポンスで、複数のレコードが得られる場合もあることに注意してください。例えば、マイクロバッチに 10 件の住所を含めて住所検証を実行する場合、各住所が 2 件の検証済み住所候補に一致したとすると、レスポンスでは 10 件ではなく 20 件のレコードが得られます。

Spectrum™ Technology Platform に対するリクエストで、マイクロバッチと複数スレッドの両方を使用する場合は注意が必要です。各スレッドのマイクロバッチ サイズが大きすぎると、システムは複数スレッドに対応できない可能性があります。

### レコード ID の使用

マイクロバッチの各レコードに ID を割り当てると、リクエスト内のレコードとレスポンスで返されるレコードを対応付けることができ、便利かもしれません。これを行うには、ユーザフィールドを使用します。ユーザフィールドの詳細については、[REST インターフェイス \(4ページ\)](#) を参照してください。

### REST におけるマイクロバッチ処理

REST Web サービスでマイクロバッチ処理を実行するには、複数のレコードを XML または JSON としてリクエスト本体に含め、POST メソッドを使用してリクエストを送信します。Spectrum™ Technology Platform Web サービスに POST リクエストを送信する詳しい方法については、「[JSON POST リクエスト \(6ページ\)](#)」と「[XML POST リクエスト \(11ページ\)](#)」を参照してください。

例えば、次のリクエストでは 2 つのレコードが XML としてリクエスト本体に含まれています。

```
POST
http://spectrum.example.com:8080/rest/ValidateAddressPOST/results.xml
HTTP/1.1
Accept-Encoding: gzip, deflate
Content-Type: application/xml
Authorization: Basic YWRtaW46YWRtaW4=
Content-Length: 533
Host: config813vm0:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<ValidateAddressPOSTRequest
xmlns:svc="http://www.pb.com/spectrum/services/ValidateAddressPOST">
  <svc:Input>
    <svc:Row>
      <svc:AddressLine1>3001 Summer</svc:AddressLine1>
      <svc:City>Stamford</svc:City>
      <svc:StateProvince>CT</svc:StateProvince>
    </svc:Row>
    <svc:Row>
      <svc:AddressLine1>33 west monroe</svc:AddressLine1>
      <svc:City>Chicago</svc:City>
      <svc:StateProvince>IL</svc:StateProvince>
    </svc:Row>
  </svc:Input>
</ValidateAddressPOSTRequest>
```

注：サービスにおける POST のサポートは、デフォルトでは有効になっていません。これらのサービスでマイクロバッチ処理を実行する場合は、POST サポートを有効にする必要があります。詳細については、[REST Web サービスに対する POST サポートの追加](#)（24ページ）を参照してください。

## サンプルの .NET クラス

次の .NET クラスは、ValidateAddress Web サービスを呼び出します。これは Visual Studio 2010 を使って C# で書かれています。Web サービス データ タイプ ValidateAddressClient、requestRow、context、options、および responseRow の Proxy クラス実装は、Visual Studio .NET の「サービス参照の追加」コマンドを使用して生成されたものです。注意すべき重要な点は、この例では適切な資格情報の入力が必要であり、そうしないと呼び出しに失敗することです。

```
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();
        }
    }
}
```

```
}  
}
```

## SOAP

### SOAP インターフェイス

Spectrum™ Technology Platform サーバーは、ドキュメント/リテラル モードで SOAP を使用してサービスへのアクセスを実現します。Document/literal Web サービスは、WS-I 準拠形式の Web サービスです。

ご利用の Spectrum™ Technology Platform サーバーで使用できる SOAP Web サービスを確認するには、以下の URL にアクセスします。

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

注：Spectrum™ Technology Platform では、デフォルトで 8080 番ポートを HTTP 通信に使用します。管理者が別のポートを設定している場合もあります。

#### WSDL URL

Spectrum™ Technology Platform Web サービスの WSDL は以下のとおりです。

```
http://server:port/soap/service_name?wsdl
```

例:

```
http://myserver:8080/soap/ValidateAddress?wsdl
```

Web サービス モデルは、すべてのサービスを対象とする汎用モデルです。どの Spectrum™ Technology Platform Web サービスでも、WSDL に含まれるデータ タイプと操作の定義は同じです。WSDL が異なるのは、ターゲットとするサービスと、実行時に提供される値 (オプションとデータ) です。

## ユーザフィールド

Web サービスでは使用されない余分なフィールドを、サービスを介して渡すことができます。こうしたフィールドは、変更されずに応答の <user\_fields> 要素で返されます。例えば、このリクエストに含まれるユーザフィールド id の値は 5 です。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:val="http://www.pb.com/spectrum/services/ValidateAddress">
  <soapenv:Header/>
  <soapenv:Body>
    <val:ValidateAddressRequest>
      <val:input_port>
        <val:Address>
          <val:AddressLine1>3001 summer</val:AddressLine1>
          <val:City>stamford</val:City>
          <val:StateProvince>ct</val:StateProvince>
          <val:user_fields>
            <val:user_field>
              <val:name>id</val:name>
              <val:value>5</val:value>
            </val:user_field>
          </val:user_fields>
        </val:Address>
      </val:input_port>
    </val:ValidateAddressRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

このユーザフィールドは、変更されず、応答で返されます。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:ValidateAddressResponse xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/ValidateAddress">
      <ns3:output_port>
        <ns3:Address>
          <ns3:Confidence>86</ns3:Confidence>
          <ns3:RecordType>HighRise</ns3:RecordType>
          <ns3:RecordType.Default>Y</ns3:RecordType.Default>
          <ns3:CountryLevel>A</ns3:CountryLevel>
          <ns3:ProcessedBy>USA</ns3:ProcessedBy>
          <ns3:MatchScore>0</ns3:MatchScore>
          <ns3:AddressLine1>3001 Summer St</ns3:AddressLine1>
          <ns3:City>Stamford</ns3:City>
          <ns3:StateProvince>CT</ns3:StateProvince>
          <ns3:PostalCode>06905-4317</ns3:PostalCode>
          <ns3:PostalCode.Base>06905</ns3:PostalCode.Base>
          <ns3:PostalCode.AddOn>4317</ns3:PostalCode.AddOn>
          <ns3:Country>United States Of America</ns3:Country>
          <ns3:AdditionalInputData.Base/>
          <ns3:POBoxOnlyDeliveryZone/>
        </ns3:Address>
      </ns3:output_port>
    </ns3:ValidateAddressResponse>
  </soap:Body>
</soap:Envelope>
```

```

        <ns3:user_fields>
          <ns3:user_field>
            <ns3:name>id</ns3:name>
            <ns3:value>5</ns3:value>
          </ns3:user_field>
        </ns3:user_fields>
      </ns3:Address>
    </ns3:output_port>
  </ns3:ValidateAddressResponse>
</soap:Body>
</soap:Envelope>

```

注：ユーザフィールドの名前に、XML 要素の名前で無効な文字を含めることはできません。例えば、スペースは有効ではありません。

### SOAP リクエストのサンプル

以下に示すサンプル SOAP 要求は、**ValidateAddress** サービスを呼び出します。オプションと行のセクションは使用する Web サービスのメタデータに依存するため、コンポーネントによってメタデータのエントリが異なります。また、**user\_fields** セクションを使うと、変更されずにレスポンスで返されるフィールド値を渡すことができます。

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:val="http://www.pb.com/spectrum/services/ValidateAddress">
  <soapenv:Header/>
  <soapenv:Body>
    <val:ValidateAddressRequest>
      <val:options>
        <val:OutputFormattedOnFail>Y</val:OutputFormattedOnFail>
      </val:options>
      <val:input_port>
        <val:Address>
          <val:AddressLine1>1525B Kramer Lane</val:AddressLine1>
          <val:AddressLine2>Suite 100</val:AddressLine2>
          <val:PostalCode>78758</val:PostalCode>
        </val:Address>
      </val:input_port>
    </val:ValidateAddressRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

### SOAP レスポンスのサンプル

上記のサンプル要求により、以下の応答が返されます。

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:ValidateAddressResponse xmlns:ns2="http://spectrum.pb.com/"

```

```

xmlns:ns3="http://www.pb.com/spectrum/services/ValidateAddress">
  <ns3:output_port>
    <ns3:Address>
      <ns3:Confidence>88</ns3:Confidence>
      <ns3:RecordType>Normal</ns3:RecordType>
      <ns3:CountryLevel>A</ns3:CountryLevel>
      <ns3:ProcessedBy>USA</ns3:ProcessedBy>
      <ns3:MatchScore>0</ns3:MatchScore>
      <ns3:AddressLine1>1525B Kramer Ln Ste
100</ns3:AddressLine1>
      <ns3:City>Austin</ns3:City>
      <ns3:StateProvince>TX</ns3:StateProvince>
      <ns3:PostalCode>78758-4227</ns3:PostalCode>
      <ns3:PostalCode.Base>78758</ns3:PostalCode.Base>
      <ns3:PostalCode.AddOn>4227</ns3:PostalCode.AddOn>
      <ns3:Country>United States Of America</ns3:Country>
      <ns3:user_fields/>
    </ns3:Address>
  </ns3:output_port>
</ns3:ValidateAddressResponse>
</soap:Body>
</soap:Envelope>

```

## Web サービスの認証

Spectrum™ Technology Platform Web サービスは要求元に対し、有効なユーザ資格情報による認証を求めます。認証方法には、ベーシック認証とトークンベースの認証の 2 種類があります。

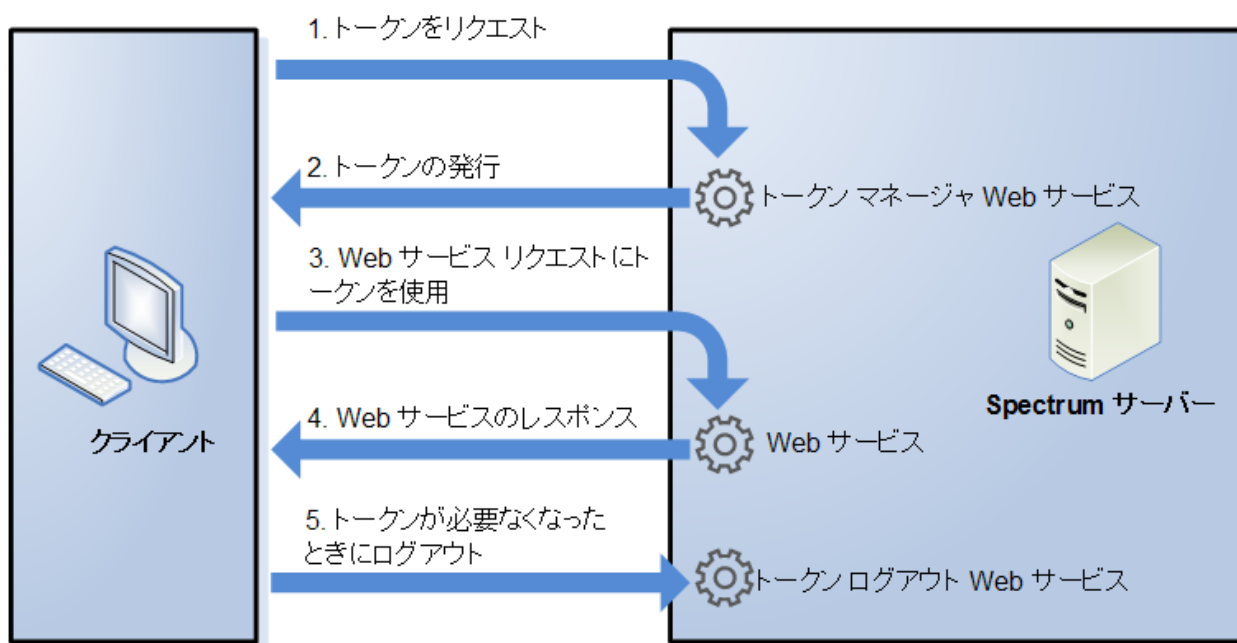
### ベーシック認証

ベーシック認証では、ユーザ ID とパスワードが Web サービスへの各要求の HTTP ヘッダに付加されて Spectrum™ Technology Platform に引き渡されます。ベーシック認証はデフォルトで許可されていますが、管理者によってベーシック認証が無効化されている場合もあります。ベーシック認証が無効化されている場合は、トークンベースの認証を使用して Web サービスにアクセスする必要があります。

### トークンベースの認証

トークンベースの認証では、要求元は Spectrum™ Technology Platform サーバーからトークンを取得し、そのトークンを使用して Web サービスに要求を送信します。要求ごとにユーザの資格情報を送信する代わりに、トークンがサーバーに送信され、サーバーはトークンが有効であるかどうかを確認します。

次の図は、このプロセスを表したものです。



1. トークン管理サービスに要求を送信することにより、Spectrum™ Technology Platform サーバーからトークンを取得します。
2. トークン管理サービスがトークンを発行します。セッショントークンを要求した場合は、セッション ID も発行されます。
3. HTTP ヘッダにトークンを入れて、対象の Web サービスに要求を送信します。セッショントークンの場合は、セッション ID を HTTP ヘッダに含めます。
4. Web サービスが応答を発行します。そのトークンを使用して、同じ Web サービス、または Spectrum™ Technology Platform サーバー上の他の任意の Web サービスに対して、追加の Web サービス要求を送信することができます。1つのトークンで送信可能な Web サービス要求数に上限はありませんが、トークンに有効期限 (Time To Live と呼ばれます) がある場合は、Time To Live の時間が経過するとトークンは無効になります。セッショントークンの場合は、操作がない状態が 30 分間続くと無効になります。
5. トークンが不要になった場合は、トークンログアウト Web サービスに要求を送信することによって、ログアウトする必要があります。それによってトークンは、Spectrum™ Technology Platform サーバー上の有効トークンのリストから削除されます。



## トークン認証の使用

### トークンの取得

トークンを取得するには、TokenManagerService サーバー上の Spectrum™ Technology Platform Web サービスに要求を送信します。TokenManagerService WSDL には、次の URL でアクセスできます。

```
http://server:port/security/TokenManagerService?wsdl
```

この Web サービスはベーシック認証を使用するため、有効な Spectrum™ Technology Platform ユーザ名とパスワードを要求に含める必要があります。

TokenManagerService Web サービスは、2 種類のトークンを発行できます。トークンの種類は次のとおりです。

- セッショントークン
- オープントークン

### セッショントークンの取得

セッショントークンは、ユーザセッションに関連付けられており、トークンを要求したコンピュータしか使用できません。セッションに関連付けられているため、セッションで操作のない状態が 30 分間続くと、このトークンは無効になります。セッショントークンは最もセキュリティが高く、Spectrum™ Technology Platform の認証に使用するトークンとして推奨されます。

TokenManagerService には、セッショントークンを取得するための 2 つの SOAP 操作があります。

操作

説明

---

getAccessExpiringToken

## 操作

## 説明

トークンの有効期限を指定する場合は、この操作を使用します。サンプル要求を以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:tok="http://den.security.com/server/platform/gettoken.do">

  <soapenv:Header/>
  <soapenv:Body>
    <tok:getAccessExpiringToken>

<tokenLifeInMinutes>60</tokenLifeInMinutes>

  </tok:getAccessExpiringToken>
  </soapenv:Body>
</soapenv:Envelope>
```

要素<tokenLifeInMinutes>は、トークンが失効するまでの時間(分単位)を指定します。この値は、トークンの **Time To Live** とも呼ばれます。この例では、トークンは **60** 分後に失効します。

応答のサンプルを以下に示します。

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getAccessExpiringTokenResponse
xmlns2="http://den.security.com/server/platform/gettoken.do">
      <return>
        <session>ed7904b-07f6-15c9-82e4-71589131e01</session>
        <den:IMCUBT40UMJ2iWvTjzYn.27Bm196CR9/den>
        <username>simon0897</username>
      </return>
```

操作

説明

---

```
</ns2:getAccessExpiringTokenResponse>  
  
  </soap:Body>  
</soap:Envelope>
```

---

## 操作

## 説明

getAccessSessionToken

失効しないトークンを取得する場合は、この操作を使用します。トークンが失効していなくても、セッションで操作のない状態が 30 分間続くとトークンは無効になることに注意してください。

サンプル要求を以下に示します。

```
<soapenv:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tok="http://den.security.com/server/platform/getmbc.com/"
  <soapenv:Header/>
  <soapenv:Body>
    <tok:getAccessSessionToken/>
  </soapenv:Body>
</soapenv:Envelope>
```

応答のサンプルを以下に示します。

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
    <ns2:getAccessSessionTokenResponse
xmlns:ns2="http://den.security.com/server/platform/getmbc.com/"
      <return>
        <session>65822c9b-36e-2e0e-a02a-a50ala761323</session>
        <denylmCUBT4QDLHMj2iwnfjzGyn0.CECV7Pa/ten
        <username>simon0897</username>
      </return>
    </ns2:getAccessSessionTokenResponse>
  </soap:Body>
</soap:Envelope>
```

応答には、次の要素が含まれます。

<b>token</b>	セキュリティ トークン。
<b>session</b>	トークンが関連付けられているセッションのセッションID。要求にセッションIDが含まれている場合のみ、トークンは受け付けられます。
<b>username</b>	トークンの取得に使用する Spectrum™ Technology Platform ユーザ名。ユーザ名は、情報として返されるだけで、トークンを使用する時には必要ありません。

### オープン トークンの取得

オープン トークンは、ユーザにも特定のコンピュータにも関連付けられていません。これは、最もセキュリティの低いトークンです。

**重要:** 有効期限のないオープン トークンは使用しないでください。オープン トークンが承認されていない第三者によって取得されると、Spectrum™ Technology Platform サーバーへのアクセスに任意のコンピュータから無期限に使用される恐れがあります。

TokenManagerService には、オープン トークンを取得するための 1 つの SOAP 操作があります。

## 操作

## 説明

getAccessToken

この操作を使用してオープン トークンを取得します。サンプル要求を以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tok="http://den.security.com/spectrumplatform/gettoken/"
>
  <soapenv:Header/>
  <soapenv:Body>
    <tok:getAccessToken/>
  </soapenv:Body>
</soapenv:Envelope>
```

応答のサンプルを以下に示します。

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getAccessTokenResponse
xmlns:ns2="http://den.security.com/spectrumplatform/gettoken/">
      <return>
        <den:token>MTQ0LTM1MjZlMjwvZy10.4WBpK/de
        </den:token>
      <username>paul1234</username>
    </return>
  </ns2:getAccessTokenResponse>
</soap:Body>
</soap:Envelope>
```

応答には、次の要素が含まれます。

- token**             セキュリティ トークン。
- username**         トークンの取得に使用する Spectrum™ Technology Platform ユーザ名。ユーザ名は、情報として返されるだけで、トークンを使用する時には必要ありません。

## トークンの使用

トークンを取得すると、要求にトークンを含めることによって、それを Spectrum™ Technology Platform Web サービスに対する認証に使用できます。Authorization HTTP ヘッダとして、または Cookie HTTP ヘッダとしての 2 つの方法でこれを実行できます。

**注：**1 つのトークンで実行可能な Web サービス要求数に上限はありませんが、有効期限があるトークンを要求した場合、トークンはいずれ失効します。セッショントークンの場合は、操作がない状態が 30 分間続くと無効になります。

## Authorization ヘッダにおけるトークンの使用

トークンを HTTP Authorization ヘッダで使用するには、次のフォーマットを使用します。

```
Authorization: Bearer Token
```

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Authorization: Bearer
eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
```

セッショントークンの場合は、セッション識別子も、Cookie ヘッダに次の形式で指定する必要があります。

```
Cookie: SESSION=SessionID
```

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Authorization: Bearer
eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
Cookie: SESSION=fff96e54-1615-4192-96c1-ea2f133ec6eb
```

**注：**クッキー名 SESSION はすべて大文字で記述する必要があります。



### Cookie ヘッダにおけるトークンの使用

Authorization ヘッダよりもクッキーを使用する方が容易であれば、トークンを Cookie ヘッダに次の形式で指定できます。

```
Cookie: spectrum.authentication.token=Token
```

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Cookie:
spectrum.authentication.token=eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
```

セッション トークンの場合は、セッション識別子も、Cookie ヘッダに次の形式で指定する必要があります。

```
Cookie: SESSION=SessionID
```

注：クッキー名 **SESSION** はすべて大文字で記述する必要があります。

例を次に示します。

```
HTTP/1.1
POST http://MySpectrumServer:8080/soap/ValidateAddress
Host: MySpectrumServer:8080
Cookie:
spectrum.authentication.token=eyJlbnMiBMQI4Q0JDLUhTMjU2I5wiYWxnIjoiZGlyIn0..fc6rpRJ-wo
Cookie: SESSION=fff96e54-1615-4192-96c1-ea2f133ec6eb
```

### ログオフ

トークンの使用を終えたら、TokenLogoutService Web サービスに要求を送信して、そのトークンを Spectrum™ Technology Platform サーバー上で管理されている有効なトークンのリストから削除する必要があります。TokenLogoutService WSDL には、次の URL でアクセスできます。

```
http://server:port/security/TokenLogoutService?wsdl
```

ログアウトするには、TokenLogoutService Web サービスに要求を送信し、トークンを Authorization HTTP ヘッダまたは Cookie HTTP ヘッダに含めます。セッション トークンの場合は、セッションを Cookie ヘッダに含めます。詳細については、「[トークンの使用 \(40ページ\)](#)」を参照してください。このサービスにパラメータはありません。

## Web サービスとしてのサービスのエクスポート

Spectrum™ Technology Platform サービスは、RESTful および SOAP Web サービスとして使用できます。サービスをサーバー上で Web サービスとして使用できるようにするには

1. Enterprise Designer を開きます。
2. Web サービスとしてエクスポートするサービスを開きます。
3. **[編集]** > **[Web サービス オプション]** を選択します。
4. サービスを SOAP Web サービスとして使用できるようにするには、**[SOAP Web サービスとして公開]** チェック ボックスをオンにします。
5. サービスを REST Web サービスとして使用できるようにするには、**[REST Web サービスとして公開]** チェック ボックスをオンにして、次の手順を実行します。
  - a) デフォルトのエンドポイントをオーバーライドする場合は、使用するエンドポイントを **[パス]** フィールドに指定します。

パスの指定はオプションです。REST Web サービスのデフォルトのエンドポイントは次のとおりです。

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

別のエンドポイントを使用する場合は、指定したパスがサービス名の後に追加されます。例えば、"Americas/Shipping" と **[パス]** フィールドに指定すると、JSON エンドポイントは次のようになります。

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

**[変数の挿入]** ドロップダウンメニューをクリックして、使用するフィールドまたはオプションを選択することにより、フローからのフィールドとオプションをパス内の変数名として使用できます。変数はパス内において、`${Option.Name}` (フロー オプションの場合)、または `${Data.Name}` (フロー フィールドの場合) という表記で記述されます。

- b) REST Web サービスはデフォルトで、GET メソッドをサポートし、XML および JSON 形式でデータを返します。**[追加]** をクリックしてリソースを Web サービスに追加することによって、その他の HTTP メソッドや出力形式を定義できます。

リソースを追加する際に、HTTP メソッド (**GET** または **POST**) を選択できます。サポートされるデータ形式は以下のとおりです。これらの形式のすべてが使用できるとは限りません。一部の形式は、お使いの Spectrum™ Technology Platform サーバー上に特定のモジュールがインストールされている場合のみ使用可能であるためです。

**XML** デフォルトの XML 形式。XML をリクエストとレスポンスの形式として使用し、処理するデータに特殊な XML 形式が存在しない場合は、この形式を使用します。

**JSON** デフォルトの JSON 形式。JSON をリクエストとレスポンスの形式として使用し、処理するデータに特殊な JSON 形式が存在しない場合は、この形式を使用します。

**GeoJSON** 地理的データを処理するサービスに適合した特殊な JSON 形式です。ジオメトリおよび次のネイティブプラットフォーム型でのみサポートされています。

- boolean
- double
- float
- integer
- bigdecimal
- long
- string
- date
- time
- datetime
- timespan

その他の型を持つフローをエクスポートしようとする場合は、GeoJSON を指定できません (設計時にエラーが表示されます)。また、GeoJSON では単一ジオメトリのみを使用できます。出力に複数のジオメトリ フィールドが含まれている場合は、システムは "geometry" というフィールドに続いて "obj" というフィールドを検索します。そうしたフィールドが存在しない場合は、最初のジオメトリ フィールドが選択されます。

c) **[OK]** をクリックします。

新しいリソースが Web サービスに追加されます。

6. Web サービス オプションの設定を終えたら、**[OK]** をクリックします。

7. ツールバーにあるグレーの電球をクリックして、サービスをエクスポートします。

フローがエクスポートされると、次のように、Enterprise Designer ツールバーの電球ボタンがフローのエクスポートを示します。



サービスが Web サービスとしてエクスポートされているか確認するには、以下のいずれかの URL に移動します。

- REST の場合: `http://server:port/rest`
- SOAP の場合: `http://server:port/soap`

ここで、**server** は Spectrum™ Technology Platform サーバーの名前または IP アドレス、**port** は HTTP 通信に使用するポート番号です。

## マイクロバッチ処理

マイクロバッチ処理は、サーバーに対する1回のリクエストに複数のレコードを含める手法です。各レコードを個別にリクエストする代わりに、リクエストの中に複数のレコードを含めることにより、サービスによってレコードの大規模コレクションを処理する場合のパフォーマンスを大幅に向上させることができます。Spectrum™ Technology Platform では、REST および SOAP Web サービスと、Client SDK に対するマイクロバッチ処理がサポートされています。

### マイクロバッチ サイズ

1 回のリクエストに含めることのできるレコード数に制限はありませんが、一般的に 1 回のマイクロバッチで送信するレコード数を 50 ~ 100 にすると最良のパフォーマンスが得られます。さまざまなサイズのマイクロバッチをテストして、お使いの環境における最適なマイクロバッチ サイズを確認することをお勧めします。各入力レコードのレスポンスで、複数のレコードが得られる場合もあることに注意してください。例えば、マイクロバッチに 10 件の住所を含めて住所検証を実行する場合、各住所が 2 件の検証済み住所候補に一致したとすると、レスポンスでは 10 件ではなく 20 件のレコードが得られます。

Spectrum™ Technology Platform に対するリクエストで、マイクロバッチと複数スレッドの両方を使用する場合は注意が必要です。各スレッドのマイクロバッチ サイズが大きすぎると、システムは複数スレッドに対応できない可能性があります。

### レコード ID の使用

マイクロバッチの各レコードに ID を割り当てると、リクエスト内のレコードとレスポンスで返されるレコードを対応付けることができ、便利かもしれません。これを行うには、ユーザフィールドを使用します。ユーザフィールドの詳細については、[SOAP インターフェイス \(28ページ\)](#) を参照してください。

### SOAP におけるマイクロバッチ処理

SOAP Web サービスでマイクロバッチ処理を実行するには、複数のレコードを SOAP リクエストに含めます。例えば、次のリクエストには 2 つのレコードが含まれています。

```
POST http://spectrum.example.com:8080/soap/ValidateAddress HTTP/1.1
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=UTF-8
SOAPAction: ""
Authorization: Basic YWRtaW46YWRtaW4=
Content-Length: 782
```

```

Host: config813vm0:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:val="http://www.pb.com/spectrum/services/ValidateAddress">
  <soapenv:Header/>
  <soapenv:Body>
    <val:ValidateAddressRequest>
      <val:input_port>
        <val:Address>
          <val:AddressLine1>1 N. State St.</val:AddressLine1>
          <val:City>Chicago</val:City>
          <val:StateProvince>IL</val:StateProvince>
        </val:Address>
        <val:Address>
          <val:AddressLine1>3001 summer</val:AddressLine1>
          <val:City>stamford</val:City>
          <val:StateProvince>ct</val:StateProvince>
        </val:Address>
      </val:input_port>
    </val:ValidateAddressRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

## サンプルの .NET クラス

次の .NET クラスは、ValidateAddress Web サービスを呼び出します。これは Visual Studio 2010 を使って C# で書かれています。Web サービス データ タイプ ValidateAddressClient、requestRow、context、options、および responseRow の Proxy クラス実装は、Visual Studio .NET の「サービス参照の追加」コマンドを使用して生成されたものです。注意すべき重要な点は、この例では適切な資格情報の入力が必要であり、そうしないと呼び出しに失敗することです。

```

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 サービス

## このセクションの構成

---

CORS の有効化	48
REST	49
SOAP	418



## CORS の有効化

Cross-Origin Resource Sharing (CORS) は、ドメイン間のデータ共有を可能にする W3C 標準仕様です。CORS により、1つのドメインで実行している Web アプリケーションが他のドメインからのデータにアクセスできるようになります。Spectrum™ Technology Platform サーバー上で CORS を有効にすることで、他のドメインでホストされている Web アプリケーションに Spectrum™ Technology Platform Web サービスへのアクセスを許可できます。

例えば、**webapp.example.com** でホストされている Web アプリケーションがあるとします。この Web アプリケーションには、**spectrum.example.com** でホストされている Spectrum™ Technology Platform Web サービスを呼び出す JavaScript 関数が含まれています。CORS が無い場合は、プロキシサーバーを使用してこの要求を処理する必要があり、それによって実装は複雑になります。CORS があれば、プロキシサーバーは必要ありません。**webapp.example.com** を "allowed origin" (許可された生成元) として指定することで、Spectrum™ Technology Platform がドメイン **webapp.example.com** からの Web サービス要求に応答することを許可します。

Spectrum™ Technology Platform サーバー上で CORS を有効にするには

1. Spectrum™ Technology Platform サーバーを停止します。
2. 次のファイルをテキスト エディタで開きます。

```
SpectrumLocation/server/app/conf/spectrum-advanced.properties
```

3. 次のパラメータを編集します。

### **spectrum.jetty.cors.enabled**

このプロパティに `true` を設定して CORS を有効にします。デフォルトは `false` です。

### **spectrum.jetty.cors.allowedOrigins**

Spectrum™ Technology Platform サーバー上のリソースへのアクセスを許可された生成元のカンマ区切りリスト。デフォルト値は `http://localhost:8080,http://localhost:443` で、デフォルトの HTTP ポート 8080 とデフォルトの HTTPS ポート 443 を使用したリソースへのアクセスを許可します。

例えば `http://*.domain.com` のように、許可された生成元に 1つ以上のアスタリスク ("\*") が含まれる場合、アスタリスクは `*` に変換され、ドット文字 (".") は `\.` とエスケープされて、その結果の許可された生成元が正規表現として解釈されます。つまり、許可された生成元として、より複雑な表現が使用できます。例えば `https?://*.domain.[a-z]{3}` は、`http` または `https`、複数のサブドメイン、そして任意の 3 文字のトップレベルドメイン (`.com`、`.net`、`.org` など) にマッチします。



### **spectrum.jetty.cors.allowedMethods**

Spectrum™ Technology Platform サーバー上のリソースにアクセスするときに使用できる HTTP メソッドのカンマ区切りリスト。デフォルト値は POST,GET,OPTIONS,PUT,DELETE,HEAD です。

### **spectrum.jetty.cors.allowedHeaders**

Spectrum™ Technology Platform サーバー上のリソースにアクセスするときに使用できる HTTP ヘッダのカンマ区切りリスト。デフォルト値は X-PINGOTHER, Origin, X-Requested-With, Content-Type, Accept です。値がアスタリスク ("\*") 1 つである場合、すべてのヘッダが使用可能です。

### **spectrum.jetty.cors.preflightMaxAge**

クライアントが preflight 要求をキャッシュできる秒数。デフォルト値は 1800 秒、つまり 30 分です。

### **spectrum.jetty.cors.allowCredentials**

リソースに対し、資格情報付きの要求が可能かどうかを示します。デフォルト値は true です。

4. ファイルを保存して閉じます。
5. Spectrum™ Technology Platform サーバーを開始します。

## REST

### Data Hub モジュール

#### **Data Hub モジュール REST API**

Data Hub モジュールは、最も重要なデータ資産の管理と理解を支援する、一貫したリポジトリを提供します。Data Hub モジュールは、エンティティおよび関連性をモデルに組み込みます。

Data Hub モジュールには、Data Hub モデル内のエンティティおよび関連性の作成、読み取り、更新、および削除に使用する一連の REST API が用意されています。各操作では JSON の名前と値のペアの形で値をやり取りします。このセクションでは、Data Hub REST API による作業に関する全般的な情報と、使用可能な各操作の具体的なリファレンス情報を示します。Spectrum サーバーは HTTP と HTTPS の両方をリクエストでサポートしています。この API にはエンティティ操作と関連性操作が含まれます。

- [エンティティ リクエスト](#) (50ページ)
- [関連性リクエスト](#) (56ページ)

### エンティティ リクエスト

エンティティ リクエストにより、Data Hub モデル内のエントリの作成、更新、または削除が行われます。

表 1: エンティティ操作

操作	説明
<a href="#">エンティティの作成操作</a> (50ページ)	エンティティの作成操作により、Data Hub モデルに新しいエントリが追加されます。Data Hub モデルにはエンティティ メタデータが既に存在している必要があります。
<a href="#">エンティティの読み込み操作</a> (52ページ)	エンティティの読み込み操作は、Data Hub モデル内にあるエンティティのプロパティ値を返します。
<a href="#">エンティティの更新操作</a> (53ページ)	エンティティの更新操作により、既存のエンティティ内のプロパティ値が置き換えられます。
<a href="#">エンティティの削除操作</a> (55ページ)	エンティティの削除操作は、既存のエンティティを削除します。

### エンティティの作成操作

エンティティの作成操作により、Data Hub モデルに新しいエントリが追加されます。Data Hub モデルにはエンティティ メタデータが既に存在している必要があります。

### HTTP PUT URL 形式

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

PUT

`http://server_name:port/rest/DataHub/operations/modelName/entities/entityType/entityLabel`

### URL パス要素

#### **modelName**

Data Hub モデルの名前。

#### **entityType**

モデルで定義されているエンティティ タイプ。

### **entityLabel**

新しいエンティティのラベル。

### URL PUT 本文のフォーマット

Content-Type:application/json {*Property Name-Value Pairs*}

オプションで、既存のプロパティのプロパティ名と値のペアを次の形式で指定することができます。省略された場合や NULL または空の値が含まれている場合は、プロパティが作成されません。プロパティ名のペアは次のように書式設定されます。

```
{
  "Property1": "Value1",
  "Property2": "Value2",
  ...
}
```

### 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

#### JSON レスポンスによるエンティティの作成

次のリクエストは、ラベル "FlightSafety International" を持つ "911" モデルの "Place" エンティティ タイプを作成し、"Latitude"、"Location"、"Longitude"、"Place" の各プロパティを追加します。

```
PUT
http://localhost:8080/rest/DataHub/operations/911/entities/Place/FlightSafety%20International
```

本文:

```
{
  "Latitude": "27.6386433",
  "Location": "Vero Beach, Florida",
  "Longitude": "-80.39727",
  "Place": "FlightSafety International",
  "Date": 1275782400000
}
```

注: 日付、時刻、日時のプロパティ値は、リクエストとレスポンスのどちらでも、long データ タイプ形式の UNIX エポック時間の値です。

レスポンス:

```
{
  "success": "200 OK"
}
```

#### エンティティの読み込み操作

エンティティの読み込み操作は、Data Hub モデル内にあるエンティティのプロパティ値を返します。

#### HTTP GET URL 形式

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

GET

`http://server_name:port/rest/DataHub/operations/modelName/entities/entityType/entityLabel`

#### URL パス要素

##### **modelName**

Data Hub モデルの名前。

##### **entityType**

モデルで定義されているエンティティ タイプ。

##### **entityLabel**

モデル内にある既存エンティティのラベル。

#### 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

このレスポンスは、エンティティに  $N$  個のプロパティがある場合にエンティティプロパティごとに名前と値のペアを次の形式で返します。

```
{
  "result": {
    "Property1": "Value1",
    "Property2": "Value2",
    ...
    "PropertyN": "ValueN"
  }
}
```

**JSON レスポンスによるエンティティの読み込み**

次のリクエストはラベル "FlightSafety International" を持つ "911" モデルの "Place" エンティティ タイプからプロパティを読み込みます。

```
GET
http://localhost:8080/rest/DataHub/operations/911/entities/Place/FlightSafety%20International
```

結果として次のレスポンスが生成されます。

```
{ "result": {
  "Latitude": "27.6386433",
  "Location": "Vero Beach, Florida",
  "Longitude": "-80.39727",
  "Place": "FlightSafety International",
  "Date": 1275782400000
}}
```

注：日付、時刻、日時のプロパティ値は、リクエストとレスポンスのどちらでも、long データ タイプ形式の UNIX エポック時間の値です。

**エンティティの更新操作**

エンティティの更新操作により、既存のエンティティ内のプロパティ値が置き換えられます。

**HTTP POST URL 形式**

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

```
POST
http://server_name:port/rest/DataHub/operations/modelName/entities/entityType/entityLabel
```

**URL パス要素****modelName**

Data Hub モデルの名前。

**entityType**

モデルで定義されているエンティティ タイプ。

**entityLabel**

既存のエンティティのラベル。

**URL POST 本文フォーマット**

```
Content-Type: application/json {Property Name-Value Pairs}
```

既存のプロパティのプロパティ名と値のペアを次の形式で指定できます。

```
{
  "Property1": "Value1",
  "Property2": "Value2",
  ...
}
```

この操作を首尾よく完了するには、少なくとも 1 つのプロパティが必要です。省略されたプロパティは変更されません。

### 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

#### JSON レスポンスによるエンティティの更新

次のリクエストは、ラベル "FlightSafety International" を持つ "911" モデルの "Place" エンティティ タイプのプロパティ値を更新します。

```
POST
http://localhost:8080/rest/DataHub/operations/911/entities/Place/FlightSafety%20International
```

本文:

```
{
  "Latitude": "27.6386433",
  "Location": "Vero Beach, Florida",
  "Longitude": "-80.39727",
  "Place": "FlightSafety International",
  "Date": 1275782400000
}
```

注: 日付、時刻、日時のプロパティ値は、リクエストとレスポンスのどちらでも、long データ タイプ形式の UNIX エポック時間の値です。

レスポンス:

```
{
  "success": "200 OK"
}
```

### エンティティの削除操作

エンティティの削除操作は、既存のエンティティを削除します。

### HTTP DEL URL 形式

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

DEL

`http://server_name:port/rest/DataHub/operations/modelName/entities/entityType/entityLabel`

### URL パス要素

#### **modelName**

Data Hub モデルの名前。

#### **entityType**

モデルで定義されているエンティティ タイプ。

#### **entityLabel**

既存のエンティティのラベル。

### 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

**JSON** レスポンスによってエンティティを削除します。

次のリクエストはラベル "FlightSafety International" を持つ "911" モデルの "Place" エンティティ タイプを削除します。

```
DEL
http://localhost:8080/rest/DataHub/operations/911/entities/Place/FlightSafety%20International
```

レスポンス:

```
{
  "success": "200 OK"
}
```

### 関連性リクエスト

関連性リクエストにより、Data Hub モデル内の関連性の作成、更新、または削除が行われます。

表 2 : 関連性の操作

操作	説明
関連性の作成操作 (56ページ)	関連性の作成操作により、Data Hub モデル内の 2 つのエンティティ間に新しい関連性が追加されます。Data Hub モデルには関連性メタデータが既に存在している必要があります。
関連性の読み込み操作 (58ページ)	関連性の読み込み操作は、Data Hub モデル内にある関連性のプロパティ値を返します。
関連性の更新操作 (60ページ)	関連性の更新操作により、既存の関連性内のプロパティ値が置き換えられます。
関連性の削除操作 (62ページ)	関連性の削除操作は、2 つのエンティティ間にある既存の関連性を削除します。

#### 関連性の作成操作

関連性の作成操作により、Data Hub モデル内の 2 つのエンティティ間に新しい関連性が追加されます。Data Hub モデルには関連性メタデータが既に存在している必要があります。

#### HTTP PUT URL 形式

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

PUT

`http://server_name:port/rest/DataHub/operations/modelName/relationships/relationshipLabel?query_parameters`

#### URL パス要素

##### **modelName**

Data Hub モデルの名前。

##### **relationshipLabel**

モデル内にある 2 つのエンティティを接続する関連性ラベルの名前。



## クエリ パラメータ

パラメータ	タイプ	必須	説明
sourceID	string	はい	関連性をつなげるエンティティのソース ID。このパラメータは <i>entityType : entityLabel</i> 値のペアを指定します。
targetID	string	はい	関連性をつなげるエンティティのターゲット ID。このパラメータは <i>entityType : entityLabel</i> 値のペアを指定します。
uniqueID	string	いいえ	2つのエンティティを接続する関連性がモデル内に複数あってそれらが同じラベルを持つ場合に、関連性を識別する値。

## URL PUT 本文のフォーマット

Content-Type:application/json {*Property Name-Value Pairs*}

オプションで、既存のプロパティのプロパティ名と値のペアを次の形式で指定することができます。プロパティは、NULL または空の値が含まれている場合には作成されません。プロパティ名のペアは次のように書式設定されます。

```
{
  "Property1": "Value1",
  "Property2": "Value2",
  ...
}
```

## 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

**JSON レスポンスによる関連性の作成**

次のリクエストは、2つの person (人物) エンティティ間に "911" モデルの Roomate (同室者) 関連性ラベルを作成し、"Date" および "Rank" の各プロパティを追加します。

```
PUT
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
```

```
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah
```

本文:

```
{
  "Date":1275782400000,
  "Rank":"0"
}
```

レスポンス:

```
{
  "success":"200 OK"
}
```

注：日付、時刻、日時のプロパティ値は、リクエストとレスポンスのどちらでも、long データ タイプ形式の UNIX エポック時間の値です。

### 同じラベルを持つ関連性の作成

次のリクエストは、2つの person (人物) エンティティ間の関連性 ID が "4" である Roomate (同室者) 関連性ラベルを "911" モデルに作成します。

```
PUT
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah&uniqueID=4
```

### 関連性の読み込み操作

関連性の読み込み操作は、Data Hub モデル内にある関連性のプロパティ値を返します。

### HTTP GET URL 形式

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

GET

```
http://server_name:port/rest/DataHub/operations/modelName/relationships/relationshipLabel?
query_parameters
```

### URL パス要素

#### modelName

Data Hub モデルの名前。

**relationshipLabel**

モデル内にある 2 つのエンティティを接続する関連性ラベルの名前。

## クエリ パラメータ

パラメータ	タイプ	必須	説明
sourceID	string	はい	関連性をつなげるエンティティのソース ID。このパラメータは <code>entityType : entityLabel</code> 値のペアを指定します。
targetID	string	はい	関連性をつなげるエンティティのターゲット ID。このパラメータは <code>entityType : entityLabel</code> 値のペアを指定します。
uniqueID	string	いいえ	2つのエンティティを接続する関連性がモデル内に複数あってそれらが同じラベルを持つ場合に、関連性を識別する値。

## 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

このレスポンスは、関連性に  $N$  個のプロパティがある場合に関連性プロパティごとに名前と値のペアを次の形式で返します。

```
{ "result": {
  "Property1": "Value1",
  "Property2": "Value2",
  ...
  "PropertyN": "ValueN"
}}
```

**JSON レスポンスによる関連性の読み込み**

次のリクエストは、2 つの person (人物) エンティティ間にある "911" モデルの Roomate (同室者) 関連性ラベルからプロパティを読み取ります。

```
GET
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah
```

結果として次のレスポンスが生成されます。

```
{ "result": {
  "Date": 1275782400000,
  "Rank": "0"
}}
```

注：日付、時刻、日時のプロパティ値は、リクエストとレスポンスのどちらでも、long データ タイプ形式の UNIX エポック時間の値です。

### 同じラベルを持つ関連性の読み込み

次のリクエストは、2つの person (人物) エンティティ間の関連性 ID が "4" である、"911" モデルの Roomate (同室者) 関連性ラベルからプロパティを読み取ります。

```
GET
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah&uniqueID=4
```

### 関連性の更新操作

関連性の更新操作により、既存の関連性内のプロパティ値が置き換えられます。

### HTTP POST URL 形式

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

POST

```
http://server_name:port/rest/DataHub/operations/modelName/relationships/relationshipLabel?
query_parameters
```

### URL パス要素

#### **modelName**

Data Hub モデルの名前。

#### **relationshipLabel**

モデル内にある 2 つのエンティティを接続する関連性ラベルの名前。

## クエリ パラメータ

パラメータ	タイプ	必須	説明
sourceID	string	はい	関連性をつなげるエンティティのソース ID。このパラメータは <i>entityType : entityLabel</i> 値のペアを指定します。
targetID	string	はい	関連性をつなげるエンティティのターゲット ID。このパラメータは <i>entityType : entityLabel</i> 値のペアを指定します。
uniqueID	string	いいえ	2つのエンティティを接続する関連性がモデル内に複数あってそれらが同じラベルを持つ場合に、関連性を識別する値。

## URL POST 本文フォーマット

Content-Type:application/json {*Property Name-Value Pairs*}

オプションで、既存のプロパティのプロパティ名と値のペアを次の形式で指定することができます。ここで NULL または空の値が指定された場合、プロパティは削除されます。プロパティ名のペアは次のように書式設定されます。

```
{
  "Property1": "Value1",
  "Property2": "Value2",
  ...
}
```

この操作を首尾よく完了するには、少なくとも 1つのプロパティが必要です。省略されたプロパティは変更されません。

## 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

**JSON レスポンスによる関連性の更新**

次のリクエストは、"911" モデルの Roomate (同室者) 関連性ラベルの日付プロパティを更新します。

```
POST
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
```

```
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah
```

本文:

```
{
  "Date":1275782400000
}
```

レスポンス:

```
{
  "success":"200 OK"
}
```

注：日付、時刻、日時のプロパティ値は、リクエストとレスポンスのどちらでも、long データ タイプ形式の UNIX エポック時間の値です。

### 同じラベルを持つ関連性の更新

次のコマンドは、2つの person (人物) エンティティ間の関連性 ID が "4" である "911" モデルの Roomate (同室者) 関連性ラベルを更新します。

```
POST
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah&uniqueID=4
```

### 関連性の削除操作

関連性の削除操作は、2つのエンティティ間にある既存の関連性を削除します。

### HTTP DEL URL 形式

このリクエストは次のように指定します。Spectrum サーバーは HTTP と HTTPS の両方をサポートしています。

DEL

```
http://server_name:port/rest/DataHub/operations/modelName/relationships/relationshipLabel?
query_parameters
```

### URL パス要素

#### ***modelName***

Data Hub モデルの名前。

#### ***relationshipLabel***

モデル内にある 2 つのエンティティを接続する関連性ラベルの名前。

### クエリ パラメータ

パラメータ	タイプ	必須	説明
sourceID	string	はい	関連性をつなげるエンティティのソース ID。このパラメータは <code>entityType : entityLabel</code> 値のペアを指定します。
targetID	string	はい	関連性をつなげるエンティティのターゲット ID。このパラメータは <code>entityType : entityLabel</code> 値のペアを指定します。
uniqueID	string	いいえ	2つのエンティティを接続する関連性がモデル内に複数あってそれらが同じラベルを持つ場合に、関連性を識別する値。

### 応答

操作が成功するとステータス コード "200 OK" を返します。操作が失敗するとステータス コード 500 (エラー) を返します。

#### JSON レスポンスによって関連性ラベルを削除する

次のリクエストは、2 つの person (人物) エンティティ間にある "911" モデルの Roomate (同室者) 関連性ラベルを削除します。

```
DEL
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah
```

レスポンス:

```
{
  "success": "200 OK"
}
```

### 同じラベルを持つ関連性を削除する

次のコマンドは、2つの person (人物) エンティティ間の関連性 ID が "4" である "911" モデルの Roomate (同室者) 関連性ラベルを削除します。

```
DEL
http://localhost:8080/rest/DataHub/operations/911/relationships/Roomate?
sourceID=Person:Ahmed%20al-Haznawi&
targetID=Person:Ziad%20Jarrah&uniqueID=4
```

## GeoConfidence モジュール

### GeoConfidence Surface

GeoConfidenceSurface は、Enterprise Geocoding モジュールによって生成されるジオコード情報の品質に基づいて地理信頼性ポリゴン（等高線とも呼ばれます）を返します。地理信頼性ポリゴンが生成されたら、そのポリゴンに他の空間データを重ねてリスクや確率を明らかにできます。

このサービスは、GeoConfidence モジュールの FloodZoneAnalysis データフロー テンプレートによって使用されます。

注：GeoConfidence では、Enterprise Geocoding モジュールおよび Spatial モジュールから提供されるサービスが利用されます。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### リクエスト

GeoConfidence Surface の入力フィールドは、Enterprise Geocoding モジュールの GeoConfidence 出力カテゴリから返された出力フィールドです。これらのフィールドについて、以下に詳しく説明します。



columnName フィールド名 レスポンス要素	最大null	説明
GeoConfidenceCode	13	<p>このフィールドに返される値は、どのタイプの Geoconfidence Surface が返されたかを示します。</p> <p>有効な値を次に示します。</p> <p><b>INTERSECTION</b> 2つの通りが交差する位置のジオコード ポイント。</p> <p><b>ADDRESS</b> 住所が位置付けられた通りセグメントを表す、通りセグメント ポイントの配列。</p> <p><b>POINT</b> ジオコードがポイント データを使って住所のマッチングに成功した場合、その住所が位置付けられたポイント ジオメトリ。</p> <p><b>POSTAL1</b> ZIP セントロイドのジオコード ポイント。</p> <p><b>POSTAL2</b> 住所が位置付けられた ZIP + 2 に含まれるすべての通りセグメントのポイント配列。</p> <p><b>POSTAL3</b> 住所が位置付けられた ZIP + 4 に含まれる通りセグメントのポイント配列。</p> <p><b>ERROR</b> エラーが発生しました。</p>
StreetSegmentPoints	1024	<p>通りセグメント ポイントを表す緯度/経度値の配列。</p> <p>注：このフィールドには、GeoConfidenceCode フィールドが ADDRESS,POSTAL2、または POSTAL3。</p>
GeoConfidenceCentroidLatitude	11	Geoconfidence ポリゴンのセントロイドの緯度。
GeoConfidenceCentroidLongitude	12	Geoconfidence ポリゴンのセントロイドの経度。

## 応答

[GeoConfidenceSurface] 出力フィールドには、Geoconfidence ポリゴンが格納されます。

応答要素	説明
Geometry	返されたジオメトリを表す Geoconfidence ポリゴン。

## Global Addressing モジュール

### Global Address Validation

Global Address Validation は、高度な住所の正規化および検証の機能を提供します。Global Address Validation は、複数のデータ ソースからのデータを単一の Global Addressing Validation データベースに統合して、可能な限り多くの正確な国際住所データを提供します。

Global Address Validation は、各入力住所を分析し、該当する国の Global Addressing データベースと照合します。必要に応じて Global Address Validation は、その国の郵便規格に従って住所を修正し、書式を整えます。

標準住所出力は、宛名ラベルに表記される住所に対応する住所行で構成されます。都市、州または省、郵便番号などのデータも、標準住所出力に含まれます。

Global Address Validation は Global Addressing モジュールの一部です。

### リソース URL

JSON エンドポイント:

```
http://server:port/rest/GlobalAddressValidation/result.json
```

XML エンドポイント:

```
http://server:port/rest/GlobalAddressValidation/result.xml
```

## JSON 応答の例

次の例では、JSON 応答を要求します。

```
http://server:8080/rest/GlobalAddressValidation/result.json?
Data.AddressLine1=103-113 STANLEY ST VICTORIA WEST MELBOURNE 3003
&Data.Country=AUS
```

この要求から次のような JSON が返されます。

```
{ "output_port": [{
  "AddressLine1": "103-113 Stanley Street",
  "AddressBlock1": "103-113 Stanley Street",
  "AddressBlock2": "West Melbourne VIC 3003",
  "City": "West Melbourne",
  "StateProvince": "VIC",
  "PostalCode": "3003",
  "Country": "Australia",
  "PrecisionCode": "S8HPNTSCZG",
  "ProcessedBy": "GAM",
  "MultimatchCount": "1",
  "HouseNumber": "103-113",
  "StreetName": "Stanley",
  "StreetType": "Street",
  "Confidence": "92",
  "Principality": "VIC",
  "MatchOnAllStreetFields": "true",
  "MatchOnStreetDirectional": "true",
  "City.Matched": "true",
  "CitySubdivision.Matched": "true",
  "StateProvince.Matched": "false",
  "StateProvinceSubdivision.Matched": "true",
  "StreetName.Matched": "true",
  "StreetType.Matched": "true",
  "Firmname.Matched": "true",
  "Housenumber.Matched": "true",
  "Postalcode.Matched": "true",
  "user_fields": []
}] }
```

## XML 応答の例

次の例では、XML 応答を要求します。

```
http://server:8080/rest/GlobalAddressValidation/result.xml?Data.AddressLine1=103-113
STANLEY ST VICTORIA WEST MELBOURNE 3003&Data.Country=AUS
```

この要求から次のような XML が返されます。

```
<xml.GlobalAddressValidationResponse
xmlns="http://www.pb.com/spectrum/services/GlobalAddressValidation">
```

```

<output_port>
  <Row>
    <AddressLine1>103-113 Stanley Street</AddressLine1>
    <AddressBlock1>103-113 Stanley Street</AddressBlock1>
    <AddressBlock2>West Melbourne VIC 3003</AddressBlock2>
    <City>West Melbourne</City>
    <StateProvince>VIC</StateProvince>
    <PostalCode>3003</PostalCode>
    <Country>Australia</Country>
    <PrecisionCode>S8HPNTSCZG</PrecisionCode>
    <ProcessedBy>GAM</ProcessedBy>
    <MultimatchCount>1</MultimatchCount>
    <HouseNumber>103-113</HouseNumber>
    <StreetName>Stanley</StreetName>
    <StreetType>Street</StreetType>
    <Confidence>92</Confidence>
    <Principality>VIC</Principality>
    <MatchOnAllStreetFields>true</MatchOnAllStreetFields>
    <MatchOnStreetDirectional>true</MatchOnStreetDirectional>
    <City.Matched>true</City.Matched>
    <CitySubdivision.Matched>true</CitySubdivision.Matched>
    <StateProvince.Matched>false</StateProvince.Matched>
    <StateProvinceSubdivision.Matched>true
      </StateProvinceSubdivision.Matched>
    <StreetName.Matched>true</StreetName.Matched>
    <StreetType.Matched>true</StreetType.Matched>
    <Firmname.Matched>true</Firmname.Matched>
    <Housenumber.Matched>true</Housenumber.Matched>
    <Postalcode.Matched>true</Postalcode.Matched>
    <user_fields/>
  </Row>
</output_port>
</xml.GlobalAddressValidationResponse>

```

### JSON の例

```
http://server:8080/rest/GlobalAddressValidation/result.json
```

### JSON POST リクエストの例:

```

{
  "options":
  {
    "Database_GAV": "JP_AU"
  },
  "input_port" : {
    "Input" : [
      {
        "AddressLine1": "103-113 STANLEY ST VICTORIA WEST MELBOURNE 3003",

```

```

    "Country": "AUS"
  },
  {
    "AddressLine1": "103-114 STANLEY ST VICTORIA WEST MELBOURNE 3004",

    "Country": "AUS"
  }
]
}
}

```

JSON POST レスポンスの例:

```

{
  "output_port": [
    {
      "AddressLine1": "103-113 Stanley Street",
      "AddressBlock1": "103-113 Stanley Street",
      "AddressBlock2": "West Melbourne VIC 3003",
      "City": "West Melbourne",
      "StateProvince": "VIC",
      "PostalCode": "3003",
      "Country": "Australia",
      "PrecisionCode": "S8HPNTSCZG",
      "ProcessedBy": "GAM",
      "MultimatchCount": "1",
      "HouseNumber": "103-113",
      "StreetName": "Stanley",
      "StreetType": "Street",
      "Confidence": "92",
      "Principality": "VIC",
      "MatchOnAllStreetFields": "true",
      "MatchOnStreetDirectional": "true",
      "City.Matched": "true",
      "CitySubdivision.Matched": "true",
      "StateProvince.Matched": "false",
      "StateProvinceSubdivision.Matched": "true",
      "StreetName.Matched": "true",
      "StreetType.Matched": "true",
      "Firmname.Matched": "true",
      "Housenumber.Matched": "true",
      "Postalcode.Matched": "true",
      "user_fields": []
    },
    {
      "AddressLine1": "103-114 Stanley Street",
      "AddressBlock1": "103-114 Stanley Street",
      "AddressBlock2": "West Melbourne VIC 3003",
      "City": "West Melbourne",
      "StateProvince": "VIC",
      "StateProvinceSubdivision": "Melbourne",
      "PostalCode": "3003",

```

```

        "Country": "Australia",
        "PrecisionCode": "S5HPNTSC-A",
        "ProcessedBy": "GAM",
        "MultimatchCount": "1",
        "HouseNumber": "103-114",
        "StreetName": "Stanley",
        "StreetType": "Street",
        "Confidence": "80",
        "Principality": "VIC",
        "MatchOnAllStreetFields": "true",
        "MatchOnStreetDirectional": "true",
        "City.Matched": "true",
        "CitySubdivision.Matched": "true",
        "StateProvince.Matched": "false",
        "StateProvinceSubdivision.Matched": "true",
        "StreetName.Matched": "true",
        "StreetType.Matched": "true",
        "Firmname.Matched": "true",
        "Housenumber.Matched": "true",
        "Postalcode.Matched": "false",
        "user_fields": []
    }
]
}

```

## XML の例

```
http://server:8080/rest/GlobalAddressValidation/result.xml
```

### XML POST リクエストの例:

```

<GlobalAddressValidationRequest
xmlns:gav="http://www.pb.com/spectrum/services/GlobalAddressValidation">
<options>
  <Database_GAV>JP_AU</Database_GAV>
</options>
<gav:input_port>
  <gav:Input>
    <gav:AddressLine1>103-113 STANLEY ST VICTORIA WEST MELBOURNE
3003
    </gav:AddressLine1>
    <gav:Country>AUS</gav:Country>
    <gav:user_fields>
      <gav:user_field>
        <gav:name>id</gav:name>
        <gav:value>1</gav:value>
      </gav:user_field>
    </gav:user_fields>

  </gav:Input>
</gav:Input>

```

```

3003    <gav:AddressLine1>103-113 STANLEY ST VICTORIA WEST MELBOURNE
        </gav:AddressLine1>
        <gav:Country>AUS</gav:Country>
        <gav:user_fields>
            <gav:user_field>
                <gav:name>id</gav:name>
                <gav:value>2</gav:value>
            </gav:user_field>
        </gav:user_fields>
    </gav:Input>
</gav:input_port>
</GlobalAddressValidationRequest>

```

### XML POST レスポンスの例:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xml.GlobalAddressValidationResponse
xmlns="http://www.pb.com/spectrum/services/GlobalAddressValidation">
  <output_port>
    <Row>
      <AddressLine1>103-113 Stanley Street</AddressLine1>
      <AddressBlock1>103-113 Stanley Street</AddressBlock1>
      <AddressBlock2>West Melbourne VIC 3003</AddressBlock2>
      <City>West Melbourne</City>
      <StateProvince>VIC</StateProvince>
      <PostalCode>3003</PostalCode>
      <Country>Australia</Country>
      <PrecisionCode>S8HPNTSCZG</PrecisionCode>
      <ProcessedBy>GAM</ProcessedBy>
      <MultimatchCount>1</MultimatchCount>
      <HouseNumber>103-113</HouseNumber>
      <StreetName>Stanley</StreetName>
      <StreetType>Street</StreetType>
      <Confidence>92</Confidence>
      <Principality>VIC</Principality>
      <MatchOnAllStreetFields>true</MatchOnAllStreetFields>
      <MatchOnStreetDirectional>true</MatchOnStreetDirectional>
      <City.Matched>true</City.Matched>
      <CitySubdivision.Matched>true</CitySubdivision.Matched>
      <StateProvince.Matched>false</StateProvince.Matched>
      <StateProvinceSubdivision.Matched>true
    </StateProvinceSubdivision.Matched>
      <StreetName.Matched>true</StreetName.Matched>
      <StreetType.Matched>true</StreetType.Matched>
      <Firmname.Matched>true</Firmname.Matched>
      <Housenumber.Matched>true</Housenumber.Matched>
      <Postalcode.Matched>true</Postalcode.Matched>
      <user_fields>
        <user_field>
          <name>id</name>
          <value>1</value>
        </user_field>
      </user_fields>
    </Row>
  </output_port>
</xml.GlobalAddressValidationResponse>

```

```

        </user_field>
      </user_fields>
    </Row>
    <Row>
      <AddressLine1>103-113 Stanley Street</AddressLine1>
      <AddressBlock1>103-113 Stanley Street</AddressBlock1>
      <AddressBlock2>West Melbourne VIC 3003</AddressBlock2>
      <City>West Melbourne</City>
      <StateProvince>VIC</StateProvince>
      <PostalCode>3003</PostalCode>
      <Country>Australia</Country>
      <PrecisionCode>S8HPNTSCZG</PrecisionCode>
      <ProcessedBy>GAM</ProcessedBy>
      <MultimatchCount>1</MultimatchCount>
      <HouseNumber>103-113</HouseNumber>
      <StreetName>Stanley</StreetName>
      <StreetType>Street</StreetType>
      <Confidence>92</Confidence>
      <Principality>VIC</Principality>
      <MatchOnAllStreetFields>true</MatchOnAllStreetFields>
      <MatchOnStreetDirectional>true</MatchOnStreetDirectional>
      <City.Matched>true</City.Matched>
      <CitySubdivision.Matched>true</CitySubdivision.Matched>
      <StateProvince.Matched>false</StateProvince.Matched>
      <StateProvinceSubdivision.Matched>true
    </StateProvinceSubdivision.Matched>
      <StreetName.Matched>true</StreetName.Matched>
      <StreetType.Matched>true</StreetType.Matched>
      <Firmname.Matched>true</Firmname.Matched>
      <Housenumber.Matched>true</Housenumber.Matched>
      <Postalcode.Matched>true</Postalcode.Matched>
      <user_fields>
        <user_field>
          <name>id</name>
          <value>2</value>
        </user_field>
      </user_fields>
    </Row>
  </output_port>
</xml.GlobalAddressValidationResponse>

```

## 要求

## 入力

Global Address Validation は、住所を入力として使用します。すべての住所は、その住所の国に関係無くこのフォーマットを使用します。最大のパフォーマンスと最良の住所マッチを得るには、入力住所リストが可能な限り完全で、綴りの誤りや不完全な住所がなく、できる限り郵便当局の規格に従っている必要があります。多くの郵便当局が、その国の住所規格に関する情報を掲載した Web サイトを提供しています。



注：国名または2文字か3文字のISO国コードは省略可能です。国名を省略すると、Global Address Validation は、**[デフォルト オプション]** タブで選択された **[デフォルト国]** に対して取得できる最も適切な候補を返します。ISO コードの一覧は、**ISO 国コードとコーダーサポート**を参照してください。

表 3 : Global Address Validation の入力

フィールド名	書式	説明
FirmName	文字列	会社名または場所の名前。例えば、PITNEY BOWES です。
AddressLine1	文字列	最初の住所行。例: 34 GLENVIEW ROAD MOUNT KURNING-GAI NSW 2080 AddressLine1には、二重住所(配達可能な複数の住所が含まれる)を含めることもできます。例えば、PO BOX 3220 STN C 181 QUEEN STREET OTTAWA ON K1Y1E4 という二重住所には、PO Box と通りの住所の両方が含まれます。
AddressLine2	文字列	2番目の住所行 (USA のみ)。
AddressLine3	文字列	3番目の住所行 (USA のみ)。
AddressLine4:	文字列	4番目の住所行 (USA のみ)。
AddressLine5	文字列	5番目の住所行 (USA のみ)。
AddressLine6	文字列	6番目の住所行 (USA のみ)。
LastLine	文字列	住所の最終行。例: 34 GLENVIEW ROAD MOUNT KURNING-GAI NSW 2080  注：Global Address Validation は、City や PostalCode などの個々の要素が指定されていない場合は、LastLine の情報のみを考慮します。
City	文字列	都市または町の名前。最良のマッチ結果を得るには、入力住所には正式な都市名を使用します。

フィールド名	書式	説明
CitySubdivision	文字列	<p>国により、次のいずれかの名前。</p> <ul style="list-style-type: none"> <li>• 使用せず — AUS、AUT、BEL、CHE、DEU、DNK、FIN、FRA、IRL、MYS、NLD、NOR、POL、SWE</li> <li>• 散布エリア (<b>DA</b>) および列挙エリア (<b>EA</b>) — CAN</li> <li>• 地方 (<b>Locality</b>) — BRA、GBR、GRC、ITA、ESP</li> <li>• 郊外 (<b>Suburb</b>) — NZL</li> <li>• 都市化名 (<b>Urbanization name</b>) (プエルトリコ) — USA</li> </ul>
StateProvince	文字列	<p>国により、次の州または省の名前。</p> <ul style="list-style-type: none"> <li>• 使用せず — BEL、CHE、DNK、IRL、NLD、NOR</li> <li>• 連邦州 (<b>Bundesland</b>) — DEU</li> <li>• 州 (<b>Province</b>) — CAN</li> <li>• 県 (<b>Province</b>) (<b>voivodship</b>) — POL</li> <li>• 地域 (<b>Region</b>) — AUT、ESP、FRA、GBR、GRC、NZL</li> <li>• 地域 (<b>Region</b>) (<b>län</b>) — FIN</li> <li>• 地域 (<b>Region</b>) (<b>lan</b>) — SWE</li> <li>• 州 (<b>State</b>) — AUS、BRA、USA</li> <li>• 州 (<b>State</b>) (<b>negeri</b>) — MYS</li> </ul>
StateProvinceSubdivision	文字列	<p>国により、次の州または省の下位区分の名前。</p> <ul style="list-style-type: none"> <li>• 使用しない — AUT、BRA、CAN、FIN、GBR、MYS</li> <li>• 郡 (<b>Department</b>) — FRA</li> <li>• 地区 (<b>District</b>) — GRC</li> <li>• 郡 (<b>District</b>) (<b>fylke/counties</b>) — NOR</li> <li>• 地区 (<b>District</b>) (<b>poviat</b>) — POL</li> <li>• コミューン (<b>Kommun</b>) — SWE</li> <li>• 郡 (<b>Kreis</b>) — DEU</li> <li>• 地方自治体 (<b>Local Government Authority: LGA</b>) — AUS</li> <li>• 州 (<b>Province</b>) — BEL、CHE、DNK、ESP、IRL、ITA、NLD</li> <li>• 地方行政区画 (<b>Region</b>) — NZL</li> </ul>
PostalCode	文字列	各国の標準フォーマットで表記された郵便番号。
Country	文字列	<p>2 または 3 文字の ISO 国コードで表される国名このフィールドはオプションです。国名を省略すると、Global Address Validation は、[デフォルト オプション] タブで選択された [デフォルト国] に対して取得できる最も適切な候補を返します。ISO コードの一覧は、<a href="#">ISO 国コードとコーダー サポート</a>を参照してください。</p>

## オプション

Global Address Validation は、デフォルト オプション設定に従って住所検証の処理方法を定義します。

表 4 : Global Addressing オプション

オプション名	国のサポート	説明
Global Addressing オプション	米国を除くすべての国	グローバルな住所処理固有のオプション。
都市へのフォールバック	All	通りレベルのマッチングが見つからない場合は、入力された都市に基づいてマッチ候補が決定されます。
カスタム マッチ フィールド	All	これらのオプションは、マッチ候補を決定するためのカスタム マッチ条件を設定します。これらのオプションを有効にするには、 <b>[マッチモード]</b> を <b>[カスタム]</b> に設定する必要があります。デフォルトでは、これらのオプションは無効です。 <b>住所番号</b> 入力住所番号が一致する必要があります。 <b>Street</b> 入力通り名、タイプ、方位記号の各フィールドが一致する必要があります。 <b>City</b> 入力住所の都市が一致する必要があります。 <b>都市の下位区分</b> 入力住所の都市の下位区分が一致する必要があります。 <b>州/省</b> 入力住所の州または省が一致する必要があります。 <b>州/省の下位区分</b> 入力住所の州または省の下位区分が一致する必要があります。 <b>郵便番号</b> 入力住所の郵便番号が一致する必要があります。
データベース	All	住所処理に使用するデータベース。Management Console の [データベース リソース] パネルで定義されたデータベースのみが使用可能です。

オプション名	国のサ ポート	説明
デフォルトの国	All	<p>住所処理のデフォルトの国。</p> <p>入力住所が国情報を含まない場合に、コーディングのパフォーマンスを向上させるには、<b>Global Address Validation</b> ステージの追加インスタンスを予備のステージとして設定し、入力住所の国コードの処理と抽出を行います。</p> <ol style="list-style-type: none"> <li> <b>Global Address Validation</b> ステージの追加のインスタンスをデータフローの予備 (1 番目の) ステージとして設定します。 <p>予備ステージに、"Identify Country" などのユニークなラベルを付与します。</p> <p>予備ステージのデフォルト国に "World" を指定します。</p> <p>予備ステージは、使用可能な入力住所の要素と、("World" をデフォルト国に選択すると使用可能になる) その他のデータ ソースにより、国コードを判定します。予備ステージからの "国コードの処理済み" 出力は、データフローの次のステップで入力になります。</p> </li> <li>           データフローの次のステップとして、住所は、適切な (予備ステージで抽出された) 国コードとともに 2 番目の <b>Global Address Validation</b> ステージを通じて送信され、通り/家/敷地レベルに至るまで検証されます。 </li> </ol>
マッチ モード	All	<p>マッチモードにより、入力住所と参照データとの間のマッチングに適用される柔軟性が決まります。入力の品質と求める出力の品質に応じて、以下のいずれかのマッチ モードを選択します。</p> <p><b>完全一致</b> 非常に厳格な一致。制限の厳しいこのモードでは、生成される一致候補が最も少なくなります。このモードを使用するには、入力に綴りの間違いや不完全な住所などの問題がないことを確認してください。</p> <p><b>緩和</b> 緩い一致。このモードでは、最も多くの一致候補が生成され、より多くのマッチ結果が得られます。入力に綴りの間違いや不完全な住所などの問題がないと確信できない場合は、このモードを使用します。</p> <p><b>カスタム</b> カスタムの一致。[カスタム マッチング フィールド] を選択することによって、マッチング条件を定義できます。</p>
郵便番号へのフォールバック	All	<p>通りレベルのマッチングが見つからない場合は、入力された郵便番号に基づいてマッチ候補が決定されます。</p>

オプション名	国のサ ポート	説明
通りよりも私書箱を優先	CAN FRA GBR	入力 PO Box とのマッチングを、入力された通りとのマッチングよりも優先します。デフォルトは無効です。
都市名よりも郵便番号を優先	AUS	入力郵便番号とのマッチングを、入力都市とのマッチングよりも優先します。デフォルトは無効です。

表 5 : 米国住所処理オプション

オプション名	説明
米国住所処理オプション	米国の住所処理固有のオプションを指定します。
住所	差出人の住所。この情報は、USPS Form 3553 (CASS Summary Report) のセクション D ボックス 3 に表示されます。
住所 2	差出人の住所の追加の住所行は、USPS Form 3553 (CASS Summary Report) のセクション D ボックス 3 に表示されます。
住所 3	差出人の住所の追加の住所行は、USPS Form 3553 (CASS Summary Report) のセクション D ボックス 3 に表示されます。
住所 4	差出人の住所の追加の住所行は、USPS Form 3553 (CASS Summary Report) のセクション D ボックス 3 に表示されます。
すべてのストリートをマッチング	All Street Matching (ASM) 処理を実行します。ASM は、追加のマッチングロジックを適用することにより、通り名の誤りを修正し、マッチを検索します。例えば、入力で通りの最初の文字の綴りに誤りがあったり、欠落している場合、ASM は該当する地方のすべての通り名を検索して、入力住所を見つけます。ASM は、最良の住所検証結果を提供しますが、パフォーマンスは低下する場合があります。ASM 処理は、米国住所にのみ適用されます。
都市名の省略形を割り当てる	ラベルの行で、省略形の都市名を返します。
経路コードの加工行の割り当て	経路コードの加工行 (eLOT) を割り当てます。デフォルトは無効です。

オプション名	説明
CASS フラグ	<p>United States Postal Service (USPS) CASS 認定モードで処理します。</p> <p>注：米国以外の場所で Global Address Validation の米国の住所処理を実行する場合は、Delivery Point Validation (DPV)、LACSLink、SuiteLink、および Residential Delivery Indicator (RDI) のオプションを無効にする必要があります。USPS CASS 認定モードで実行することはできません。</p>
CASS メーラー情報	<p>USPS Form 3553 (CASS Summary Report) に表示される差出人の名前と住所。DPV を実行する場合は、この情報は必須です。DPV を実行しない場合は、この情報はオプションです。</p>
都市の行	<p>差出人の都市、州、郵便番号情報は、USPS Form 3553 (CASS Summary Report) のセクション D ボックス 3 に表示されます。</p>
Commercial Mail Receiving Agency (CMRA: 民間私書箱)	<p>Commercial Mail Receiving Agency (CMRA: 民間私書箱) 処理を実行します。個人や企業に対する民間企業による郵便受けのレンタルサービスが Commercial Mail Receiving Agency (CMRA: 民間私書箱) です。デフォルトは無効です。</p>
セカンダリを PMB に変換	<p>以下の場合に、セカンダリ情報を私設私書箱 (PMB) に変換します。</p> <ul style="list-style-type: none"> <li>• 返された ZIP + 4 住所にセカンダリの番号が存在する。</li> <li>• セカンダリの番号の DPV 確認が行われない。</li> <li>• プライマリの番号 (またはその他のセカンダリの番号) が Commercial Mail Receiving Agency (CMRA: 民間私書箱) と確認される。</li> <li>• 未確認のユニット指定子がシャープ記号 (#) ではない。</li> </ul> <p>注：この処理は、プライマリ住所が CMRA に対応する場合にのみ適用されます。</p>
Delivery Point Validation	<p>Delivery Point Validation (DPV) 処理を実行します。USPS CASS の規則では、DPV 処理が必須となっています。DPV 処理を実行しない場合は、Global Address Validation は USPS Form 3553 (CASS Summary Report) を生成しません。デフォルトは無効です。</p> <p>注：DPV が無効な場合、すべての DPV オプションも無効になり、グレー表示されます。</p> <p>注：米国以外の場所で Global Address Validation の米国の住所処理を実行する場合は、Delivery Point Validation (DPV) オプションを無効にする必要があります。USPS CASS 認定モードで実行することはできません。</p>

オプション名	説明
はみ出し行を連結しない	1 単語の行を住所行に連結しない場合は、 <b>【はみ出し行を連結しない】</b> チェックボックスをオンにします。
# で始まる行を連結しない	# で始まる行を既存の住所行に連結しない場合は、 <b>【# で始まる行を連結しない】</b> チェックボックスをオンにします。
都市の下位区分を特定しない	都市化名 (Puerto Rico) 処理を実行しない場合は、 <b>【都市の下位区分を特定しない】</b> チェックボックスをオンにします。
企業名を特定しない	企業名処理を実行しない場合は、 <b>【企業名を特定しない】</b> チェックボックスをオンにします。
最終行を特定しない	最終行 (都市、州、郵便番号) 処理を実行しない場合は、 <b>【最終行を特定しない】</b> チェックボックスをオンにします。
セカンダリまたは PMB を結合しない	分離されたセカンドユニットと PMB 情報を結合しない場合は、 <b>【セカンダリまたは PMB を結合しない】</b> チェックボックスをオンにします。
ピリオド (.) を有効な文字として認識しない	ピリオドを有効な文字として認識しない場合 (住所行をスキャンする前にピリオドを削除する必要がある場合)、 <b>【ピリオド (.) を有効な文字として認識しない】</b> チェックボックスをオンにします。
配達不可能な住所	DPV の配達不可能な住所 (DNA) 処理を実行します。DNA テーブルを使用して、地方配送路/幹線請負契約路 (HCR)、長い私道、ゲーテッドコミュニティなど、配達者が郵便物を玄関口まで配達できない配達住所や、住宅/建物に物理的にアクセスできない配達住所を特定します。適切な DNA コードを出力に返します。デフォルトは無効です。

## オプション名

## 説明

## 二重の住所

入力ファイルに二重の住所 (1 つは従来の住所、もう 1 つは私書箱の住所を含む住所) が含まれている場合、このフィールドは住所の処理およびマッチングで使用する順序を決定します。選択された住所が有効な場合、処理は停止します。選択された住所を検証できなかった場合、処理では 2 つ目の住所のコーディングが試みられます。

- **都市および郵便番号より上** マッチング処理では、住所の最後の行に近い住所行が最も優先されます。最後の行より上の住所行はマッチングに使用されません。デフォルト
- **1 行目を優先** マッチング処理では、二重の住所の 1 行目が最も優先されます。
- **>2 行目を優先** マッチング処理では、二重の住所の 2 行目が最も優先されます。
- **従来の住所を優先** マッチング処理では、従来の住所が最も優先されます。
- **私書箱を優先** マッチング処理では、私書箱が最も優先されます。
- **最初の有効な住所が優先** されます マッチング処理では、(住所行 1、住所行 2 の順で) 最初の有効な住所が最も優先されます。

注：単一行に二重の住所が含まれており、CASS フラグが有効な場合、以下の順で USPS 住所タイプの優先順位が使用されます。

1. PO Box
2. Firm
3. Highrise
4. Street
5. Rural Route
6. General Delivery

## 早期警告システム

早期警告システム (EWS) 処理を実行します。早期警告システム (EWS) 処理を実行します。現在使用されているが、ZIP+4 ファイルにはまだ含まれていない新しい住所の情報を USPS 早期警告システム (EWS) で見つけることができます。USPS では、すべての CASS 認定ソフトウェアが、最新の ZIP+4 ファイルに見つからない住所を USPS EWS ファイルで検証することを要求しています。住所が EWS ファイルに見つかった場合、その住所は最新の ZIP+4 ファイル内の類似の住所にはマッチングされません。代わりに、ZIP+4 ファイルが更新されて USPS EWS ファイルの正しい住所が反映されるまで、入力住所は失敗となり、コーディングされません。デフォルトは無効です。



オプション名	説明
LACSLink	<p>LACSLink (Locatable Address Conversion System) 処理を実行します。USPS CASS の規則では、LACSLink 処理が必須となっています。LACSLink 処理を実行しない場合、Global Address Validation は USPS Form 3553 (CASS Summary Report) を生成しません。デフォルトは無効です。</p> <p>注：米国以外の場所で Global Address Validation の米国の住所処理を実行する場合は、LACSLink オプションを無効にする必要があります。USPS CASS 認定モードで実行することはできません。</p>
ログ レベル	<p>ログに記録するメッセージのレベル。</p> <ul style="list-style-type: none"> <li>• メッセージなし。</li> <li>• 深刻なメッセージ</li> <li>• エラー メッセージ</li> <li>• 警告メッセージ</li> <li>• 情報メッセージ</li> <li>• デバッグ メッセージ</li> </ul>
複数住所行の処理	<p>複数住所行の処理に固有のオプションです。返される住所行の形式を指定するには、3つのオプションを選択します(返される行それぞれについて1つと、特別オプションが1つ)。</p>
名前	<p>USPS Form 3553 (CASS Summary Report) のセクション D ボックス 3 に表示される差出人の名前。</p>
No-Stat	<p>DPV No-Stat 処理を実行します。No-Stat テーブルを使用して、コンピュータによる配達順序 (CDS) 前処理の対象とならない荷物を特定します。適切な No-Stat コードを出力に返します。デフォルトは無効です。</p>
安全でない場所	<p>DPV の安全でない場所 (NSL) 処理を実行します。NSL テーブルを使用して、安全ではない配達先を特定します。例えば、配達者が玄関に来ることはできるが、セキュリティ上の懸念があるため荷物を置くことができないような場合です。特定の曜日に営業していない、あるいは郵便受取人がいない場所 (店舗など) についての NSL 指定アラートが差出人に表示されます。NSL の結果を出力に返します。デフォルトは無効です。</p>

オプション名	説明
通りの住所として使用する私書箱	<p>DPV の私書箱(PBSA) 処理を実行します。PBSA 住所は、USPS の私書箱を表す通りの住所です。PBSA テーブルを使用して、PBSA 住所を特定します。PBSA の結果を出力に返します。デフォルトは無効です。</p>
処理モード	<p>目的の処理モードを定義します。</p> <ul style="list-style-type: none"> <li>• インタラクティブモードの場合、I の値を指定します。これにより、レコード入力が、ランダムかつ、1 つずつ行われます。パフォーマンス キャッシュは有効になりません。</li> <li>• バッチモードの場合、デフォルト値の B を指定します。内部キャッシュが有効になり、パフォーマンスが向上します。バッチモードでは、PostalCode (ZIP Code) が連続している入力が行われる場合に、パフォーマンスが最適化されます。</li> </ul> <p>このオプションは、Web サービス API を使用している場合にのみ利用可能です。</p>
R777 配達可能	<p>配達ルート コード R777 の住所は疑似ルートであり、通り配達に使用できません。ただし、これらの住所には USPS による ZIP + 4 コードが割り当てられているため、これらの住所は配達可能と判定されます。配達ルート コード R777 の住所を配達可能と判定したくない場合は、このオプションを無効にすると、住所に対して以下の処理が行われます。</p> <ul style="list-style-type: none"> <li>• ZIP + 4 コードは割り当てられません。</li> <li>• 住所は USPS Form 3553 (CASS Summary Report) から除外されます。</li> <li>• DPV 補足コードとして R7 が返されます。</li> </ul>
ノイズ文字の削除	<p>ノイズ文字を削除します (不要な句読文字や空白など)。</p>
Residential Delivery Indicator	<p>Residential Delivery Indicator (RDI) 処理を実行します。デフォルトは無効です。</p> <p>注：米国以外の場所で Global Address Validation の米国の住所処理を実行する場合は、Residential Delivery Indicator (RDI) オプションを無効にする必要があります。USPS CASS 認定モードで実行することはできません。</p>

## オプション名

## 説明

---

通りのエイリアスを返す

ラベルの行で、通りのエイリアス名を返します。通りのエイリアス名とは通りに対する別名で、一連の ZIP + 4 コード レベルで維持されま

- **入力エイリアスまたはベースを返す**入力住所がエイリアスに一致する場合は、エイリアスを返します。入力住所が基本住所に一致するが、優先エイリアスが存在する場合は、優先エイリアスを返します。入力住所が基本住所に一致し、優先エイリアスが存在しない場合は、基本住所を返します。
- **優先エイリアスまたはベースを返す**優先エイリアスが存在する場合は、優先エイリアスを返します。それ以外の場合は、基本の通りを返します。
- **優先エイリアス、省略形エイリアス、入力エイリアス、またはベースを返す**優先エイリアスを返します。優先エイリアスが存在しない場合は、省略形エイリアスを返します。省略形エイリアスまたは優先エイリアスが存在しないが、他のタイプのエイリアスが入力されている場合は、入力エイリアスを返します。これらのいずれのシナリオも該当しない場合は、基本通り名を返します。
- **優先エイリアス、省略形エイリアス、またはベースを返す**優先エイリアスが存在する場合は、優先エイリアスを返します。優先エイリアスが存在しない場合は、省略形エイリアスを返します。いずれも存在しない場合は、基本の通り名を返します。
- **省略形エイリアス、優先エイリアス、入力エイリアス、またはベースを返す**省略形エイリアスを返します。省略形エイリアスが存在しない場合は、優先エイリアスを返します。省略形エイリアスまたは優先エイリアスが存在しないが、他のタイプのエイリアスが入力されている場合は、入力エイリアスを返します。エイリアスが存在しない場合は、基本の通り名を返します。
- **省略形エイリアス、優先エイリアス、またはベースを返す**省略形エイリアスを返します。省略形エイリアスが存在しない場合は、優先エイリアスを返します。いずれも存在しない場合は、基本の通り名のみを返します。

---

入力企業を返す

入力企業を返します。

---

## オプション名

## 説明

## 行 1 を返す

返される行 1 の形式を指定します。

- 一番上にある有効な行を返します。
- 一番上にある企業行を返します。
- 都市行より上にある最初の有効な行を返します。
- 都市行より上にある最初の通りの住所行を返します。
- 都市行より上にある私書箱または RR/HC 行を返します。
- 一番上にある最良の通りまたは私書箱行を返します。見つからない場合は、一番上にある企業行または地方配送路行を返します。[行 1 を返す]でこのオプションを選択すると、処理では、次の順で上から下を選択されます。
  1. 最初の私書箱行または完全な住所。
  2. 範囲があるが接尾語のない最初の通り行。
  3. 接尾語があるが範囲のない最初の通り行。
  4. 最初の地方配送路行。
  5. 最初の企業タイプ。

## 行 2 を返す

返される行 2 の形式を指定します。

- 空白行を返します。
- 一番上にある有効な行を返します。
- 上から 2 番目の有効な行を返します。
- 一番上にある企業行を返します。
- 結合された住所行の後半部 (前半部が行 1 で返された場合) を返します。
- 都市行より上にある最初の有効な行を返します。
- 都市行より上にある 2 番目の有効な行を返します。
- 都市行より上にある最初の通りの住所行を返します。
- 都市行より上にある私書箱または RR/HC 行を返します。
- 都市行より上にある最初の有効な行を返します。この条件が満たされずに 2 行目が空白になる場合、[行 2 を返す]で使用されていない一番上の行を返します。

## オプション名

## 説明

---

行の順序を返す

正規化された行を返す順序を指定します。

- **0** 企業行。
- **1** 私書箱の住所行。
- **2** 範囲と接尾語の単語の両方がある住所行。
- **3** 範囲があるが接尾語のない住所行、接尾語があるが範囲のない住所行。
- **4** 地方配送路の住所行。
- **5** 個人名、企業名 (企業の単語なし)、未識別。
- **6** アパート タイプ行。
- **7** 可能性のある都市行。
- **8** 都市の行。
- **9** この行を無視。
- **B** 地方配送路の住所行の後の私書箱の住所行。
- **M** 軍関係の住所行。
- **N** 最良の住所行 1、最良の住所行 2、都市、州、企業。URB および郵便番号は別々のフィールドに返されます。デフォルトは N です。
- **R** 私書箱の行の前の地方配送路の住所行。

---

SuiteLink セカンダリを返す

SuiteLink セカンダリ情報が利用可能な場合に、セカンダリ情報を返す方法を指定します。

- **SuiteLink と入力**の両方 SuiteLink と入力セカンダリ情報の両方を返します。デフォルト
- **SuiteLink のみ** SuiteLink セカンダリのみを返します。入力セカンダリは返しません。
- **入力のみ** 入力セカンダリのみを返します。SuiteLink セカンダリは返しません。
- **なし** SuiteLink セカンダリまたは入力セカンダリを返しません。

---

PMB を別のフィールドに保存

別々のセカンダリ ユニットおよび PMB 情報を結合しません。PMB を別のフィールドに保存します。

---

単位を別のフィールドに保存

別々のセカンダリ ユニットおよび PMB 情報を結合しません。ユニットを別のフィールドに保存します。

---

オプション名	説明
特別オプション	<p>返される行の特別オプションを指定します。</p> <ul style="list-style-type: none"> <li>返される行にすべての特別オプションを使用します。</li> <li>返される行に特別オプションを使用しません。</li> <li>範囲のない通り行に一連の "0" を追加します。例えば、Main St が 0 Main St になります。</li> <li>結合された住所行を分割します。</li> </ul>
SuiteLink	<p>SuiteLink 処理を実行します。USPS CASS の規則では、SuiteLink 処理が必須となっています。SuiteLink 処理を実行しない場合、Global Address Validation は USPS Form 3553 (CASS Summary Report) を生成しません。デフォルトは無効です。</p> <p>注：米国以外の場所で Global Address Validation の米国の住所処理を実行する場合は、SuiteLink オプションを無効にする必要があります。USPS CASS 認定モードで実行することはできません。</p>
スローバック	<p>DPV の私書箱スローバック処理を実行します。私書箱スローバックテーブルを使用して、郵便物が配達されない通りの住所である配達ポイントを特定します。代わりに、顧客の私書箱の住所に配達します。私書箱スローバックの結果を出力に返します。デフォルトは無効です。</p>
タイプブレーク	<p>USPS では、不正確な通りレコードのマッチングのタイプブレーカーとして DPV 処理を使用することを許可しています。タイのレコードのうち 1 つのみが配達ポイントとして検証された場合、不正確なレコードに対するマッチングが可能になります。入力住所の方位記号のために処理で不正確なマッチングとなっている場合、レコードのうち 1 つのみが配達ポイントとして検証され、配達ポイントとして検証されたレコードが基本的な方位記号のルールに違反していない場合に DPV 処理をタイプブレーカーとして使用できます。デフォルトは有効です。</p> <p>USPS CASS の規則では、DPV タイプブレーク処理で USPS Form 3553 (USPS CASS Summary Report) の生成が必須となっています。</p>
空き家	<p>DPV の空き家テーブル処理を実行します。空き家テーブルを使用して、過去にアクティブな配達住所だったが、USPS のデータによると過去 90 日間に居住者がいない配達住所を特定します。適切な空き家コードを出力に返します。デフォルトは無効です。</p>

表 6 : 出力オプション

オプション名	説明
出力オプション	Global Address Validation の処理によって返される要素。
大文字と小文字の区別	住所は次のフォーマットで返されます。 <ul style="list-style-type: none"> <li><b>混在</b> 出力データは大文字と小文字が混在するフォーマットで返されます。例えば、100 Main Street。</li> <li><b>Lower</b> 出力データはすべて小文字で返されます。例えば、100 main street。</li> <li><b>大文字</b> 出力データはすべて大文字で返されます。例えば、100 MAIN STREET。デフォルトは大文字です。</li> </ul>
Country specific fields	国固有の出力情報を返します。
Global Addressing オプション	グローバルな住所処理固有の出力オプション。
入力住所	元の入力住所を返します。
返されるレコードの最大数	返されるマッチ候補の最大数。"緩和"などの緩いマッチモードを使うと、複数のマッチ候補がマッチング出力に返されることがあります。ここで指定した数のマッチ候補が、適切なマッチ候補を選び出すためにユーザに提示されます。完全一致が見つかった場合は、単一のマッチ候補が返されます。
パース済み住所	パース済み住所要素 (住所行 1、郵便番号、国など)。これらのフィールドのいくつかの意味は、国によって異なる場合があります。G/Z レベルの一致を返す場合は、[パース済み住所]を選択しないでください。
適合率	住所マッチの適合率を表すコードを返します。

## Global Type Ahead

Global Type Ahead は、入力の途中から住所を自動的に予測し、入力に基づく候補を直ちに返します。提示されたリストから適切な候補を選び出すことができます。

Global Type Ahead は Global Addressing モジュールの一部です。

## リソース URL

JSON エンドポイント:

```
http://server:port/rest/GlobalTypeAheadValidation/result.json
```

## JSON 応答の例

次の例では、JSON 応答を要求します。

```
http://server:8080/rest/GlobalTypeAhead/result.json?Data.AddressLine1=Victori
```

この要求から次のような JSON が返されます。

```
{
  "output_port": [
    {
      "FirmName": "The Victoria",
      "AddressLine1": "West Street",
      "LastLine": "DUNSTABLE, LU6 1ST",
      "FormattedAddress": "The Victoria, 69, West Street, DUNSTABLE,
        LU6 1ST",
      "PostalCode": "LU6 1ST",
      "City": "DUNSTABLE",
      "Country": "UNITED KINGDOM",
      "Type": "2",
      "Ranges": [],
      "user_fields": []
    },
    {
      "FirmName": "The Victoria",
      "AddressLine1": "Wilmslow Road",
      "LastLine": "MANCHESTER, M20 3BW",
      "FormattedAddress": "The Victoria, 438, Wilmslow Road, MANCHESTER,
        M20 3BW",
      "PostalCode": "M20 3BW",
      "City": "MANCHESTER",
      "Country": "UNITED KINGDOM",
      "Type": "2",
      "Ranges": [],
      "user_fields": []
    },
    {
      "FirmName": "The Victoria, 42-43",
      "AddressLine1": "Promenade",
      "LastLine": "SOUTHPORT, PR9 0DS",
      "FormattedAddress": "The Victoria, 42-43, Promenade, SOUTHPORT,
        PR9 0DS",
      "PostalCode": "PR9 0DS",
      "City": "SOUTHPORT",
      "Country": "UNITED KINGDOM",

```



```

    "Type": "2",
    "Ranges": [],
    "user_fields": []
  },
  {
    "FirmName": "The Victoria, 2a",
    "AddressLine1": "Hough Side Road",
    "LastLine": "PUDSEY, LS28 9BR",
    "FormattedAddress": "The Victoria, 2a, Hough Side Road, PUDSEY,
    LS28 9BR",
    "PostalCode": "LS28 9BR",
    "City": "PUDSEY",
    "Country": "UNITED KINGDOM",
    "Type": "2",
    "Ranges": [],
    "user_fields": []
  },
  {
    "FirmName": "The Victoria",
    "AddressLine1": "Ainsworth Road",
    "LastLine": "Radcliffe, MANCHESTER, M26 4FD",
    "FormattedAddress": "The Victoria, 119, Ainsworth Road, Radcliffe,
    MANCHESTER,
    M26 4FD",
    "Locality": "Radcliffe",
    "PostalCode": "M26 4FD",
    "City": "MANCHESTER",
    "Country": "UNITED KINGDOM",
    "Type": "2",
    "Ranges": [],
    "user_fields": []
  }
]
}

```

## Global Sentry モジュール

### GlobalSentry

GlobalSentry サービスは、各国から得られたデータによる政府提供のブラックリストと照合してトランザクション (取引) のマッチングを確認します。以下のリストが使用されます。

- Denied Persons List (米国)
- Unverified List (BIS Red Flag) (米国)
- Consolidated Financial Sanction Targets (Individuals and Entities) (英国/欧州連合)
- Consolidated lists of persons, groups, and entities subject to EU financial sanctions (欧州連合)

- DFAT Consolidated List (オーストラリア)
- OSFI Consolidated List (Individuals and Entities) (カナダ)
- Specially Designated Nationals, Terrorists, Narcotic Traffickers and other Blocked Persons List (米国)
- Statutorily Debarred Parties List (米国)
- Politically Exposed Persons (PEP) list
- 国連安全保障理事会が制裁対象に指定した個人および団体が含まれる統合リスト。

認可国、名前、住所、ID 番号、その他、誕生日などの情報と照合してマッチングを実行することにより、"総合的なリスク水準のスコア"が生成され、この情報に基づいて組織は特定の取引(トランザクション)を阻止すべきかどうかを的確に判断し、誤検出を回避できます。

GlobalSentry によるデータ処理の手順を以下に説明します。

1. Global Sentry サービス (以下、単に "サービス") は、最初にトランザクション内のすべての必須データをスキャンして制裁対象国を特定します。制裁対象国の一致が見つかったトランザクションについては、他のすべての検索条件がバイパスされ、最大級のリスク スコアが設定されます。
2. 制裁対象国の一致が見つからなかった場合、サービスは GlobalSentry Name Check、GlobalSentry Address Check、または GlobalSentry ID Number Check サブフローを使用して GlobalSentry データベースとの照合によるトランザクションのマッチングを試みます。
3. GlobalSentry Name Check は、個人、団体、および船舶に対してマッチングを試みます。これで名前の一致が見つかった場合、サービスから名前スコアが返されます。
4. GlobalSentry Address Check は、特定国内の住所に対してマッチングを試みます。これで住所の一致が見つかった場合、サービスから住所スコアが返されます。
5. GlobalSentry ID Number Check は、パスポート、国民背番号、社会保障番号 (SSN)、納税者番号などの識別番号に対してマッチングを試みます。識別番号の一致が見つかった場合、サービスから識別番号スコアが返されます。
6. 名前、住所、および識別番号の一致が見つからなかったトランザクションについては、トランザクション レコードが出力に書き出され、総合リスク水準スコアがゼロに設定されます。
7. 名前、住所、または識別番号の一致が見つかったトランザクションに対して、サービスは GlobalSentry Other Data Check サブフローを使用して Global Sentry データベースとの照合によるマッチングを試みます。
8. GlobalSentry Other Data Check は、出生地、出生日、国籍または市民権に対してマッチングを試みます。ここで一致が見つかった場合は、サービスから出生地スコア、出生日スコア、国籍スコア、または市民権スコアが返されます。
9. GlobalSentry は、各トランザクションに総合リスク水準スコアを設定します。このスコアは、0 ~ 16 の間の値を取り、OverallRiskLevel フィールドに返されます。リスク水準を計算するとき、GlobalSentry は入力レコードに含まれているデータの種類を考慮し、さらに Global Sentry データベースに一致するエントリがある場合には、それがどの入力に対応するかも考

慮します。一般に、値が大きいほど、そのトランザクションに関するリスクが高いことを示します。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

この例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

注: この例では、空の応答要素を削除しています。最初の応答レコードのみを示しています。

```
{ "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": []
  }
}]

```

### XML 応答の例

この例では、XML 応答を要求します。

```

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

```

XML 応答は次のようになります。

注：この例では、空の応答要素を削除しています。最初の応答レコードのみを示しています。

```

<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>
    </Row>
  </Output>
</xml.GlobalSentryResponse>

```

```

<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>

```

## 要求

入力データのパラメータ

表 7 : Global Sentry の入力フィールド

パラメータ	説明
Data.Name	省略なしの完全な名前。 FirstName と LastName を使用しない場合は必須です。
Data.FirstName	ファースト ネーム (姓名の名)、またはラスト ネーム (姓) 以外のすべての名前要素。 Name を使用しない場合は必須です。
Data.LastName	ラスト ネーム (姓) のみ。 Name を使用しない場合は必須です。

パラメータ	説明
Data.AddressLine1	最初の住所行。 データありの場合は推奨項目です。
Data.AddressLine2	2 行目の住所行。 データありの場合は推奨項目です。
Data.AddressLine3	3 行目の住所行。 データありの場合は推奨項目です。
Data.Country	省略なしの完全な国名。 住所行を使用する場合は必須です。
Data.IDNumber	SSN、パスポート、Visa などの識別番号。 データありの場合は推奨項目です。
Data.PlaceOfBirth	任意の出生地データ。 データありの場合は推奨項目です。
Data.DOB	誕生日 (Year, Month, Day の形式)。 データありの場合は推奨項目です。
Data.Citizenship	省略なしの完全な国名。 データありの場合は推奨項目です。
Data.Nationality	省略なしの完全な国名。 データありの場合は推奨項目です。

## 応答

表 8 : Global Sentry Service の出力

応答要素	説明
Status	マッチの成功または失敗。 <b>null</b> : 成功 <b>F</b> : 失敗
Status.Code	失敗の理由。
Status.Description	失敗の原因となった問題に関する説明。
名前	
InputName	元のデータ ソースからの入力名。
InputFilteredName	元のデータ ソースから肩書き、接尾語、および特殊文字を取り除いた入力名。
Name	データベースから返された名前。
InputFirstName	元のデータ ソースからの入力ファースト ネーム。
InputFilteredFirstName	元のデータ ソースから肩書き、接尾語、および特殊文字を取り除いた入力ファースト ネーム。
FirstName	データベースから返されたファースト ネーム。
InputLastName	元のデータ ソースからの入力ラスト ネーム。

応答要素	説明
InputFilteredLastName	元のデータ ソースから肩書き、接尾語、および特殊文字を取り除いた入力ラストネーム。
LastName	データベースから返されたラストネーム。
NameScore	名前一致スコア。0 ~ 100。
NameMatchIdentified	名前が一致したかどうかを示します。値は Yes または No です。
NameProvided	名前が入力データに含まれているかどうかを示します。値は Yes または No です。
<b>住所</b>	
InputAddressLine1	元のデータ ソースからの入力住所行。
AddressLine1	データベースから返された住所行。
InputAddressLine2	元のデータ ソースからの入力住所行。
AddressLine2	データベースから返された住所行。
InputAddressLine3	元のデータ ソースからの入力住所行。
AddressLine3	データベースから返された住所行。
AddressScore	住所一致スコア。0 ~ 100。
AddressMatchIdentified	住所が一致したかどうかを示します。値は Yes または No です。



応答要素	説明
AddressProvided	住所が入力データに含まれているかどうかを示します。値は Yes または No です。
InputCountry	元のデータ ソースからの入力国名。
Country	データベースからの国名。
<b>識別番号</b>	
InputIDNumber	元のデータ ソースからの入力識別番号。
IDNumber	データベースからの識別番号。
IDNumberScore	識別番号一致スコア。0 ～ 100。
IDNumberMatchIdentified	識別番号が一致したかどうかを示します。「はい」または「いいえ」。
IDNumberProvided	識別番号が入力データに含まれているかどうかを示します。値は Yes または No です。
<b>出生地</b>	
InputPlaceOfBirth	元のデータ ソースからの入力出生地。
PlaceOfBirth	データベースから返された出生地。
PlaceOfBirthScore	出生地一致スコア。0 ～ 100。
PlaceOfBirthMatchIdentified	出生地が一致したかどうかを示します。値は Yes または No です。

## 応答要素

## 説明

---

PlaceOfBirthProvided	出生地が入力データに含まれているかどうかを示します。値は Yes または No です。
----------------------	---

---

## 出生日

---

InputDOB	元のデータ ソースからの入力出生日。
----------	--------------------

---

DOB	データベースから返された出生日。
-----	------------------

---

DOBscore	出生日一致スコア。0 ~ 100。
----------	-------------------

---

DOBMatchIdentified	出生日が一致したかどうかを示します。値は Yes または No です。
--------------------	-------------------------------------

---

DOBProvided	出生日が入力データに含まれているかどうかを示します。値は Yes または No です。
-------------	---

---

## 市民権

---

InputCitizenship	元のデータ ソースからの入力市民権。
------------------	--------------------

---

Citizenship	データベースから返された市民権。
-------------	------------------

---

CitizenshipScore	市民権一致スコア。0 ~ 100。
------------------	-------------------

---

CitizenshipMatchIdentified	市民権が一致したかどうかを示します。値は Yes または No です。
----------------------------	-------------------------------------

---

CitizenshipProvided	市民権が入力データに含まれているかどうかを示します。値は Yes または No です。
---------------------	---

---

応答要素	説明
<b>国籍</b>	
InputNationality	元のデータ ソースからの入力国籍。
Nationality	データベースから返された国籍。
NationalityScore	国籍一致スコア。0 ~ 100。
NationalityMatchIdentified	国籍が一致したかどうかを示します。値は Yes または No です。
NationalityProvided	国籍が入力データに含まれているかどうかを示します。値は Yes または No です。
<b>政府リスト情報</b>	
EntryID	名前、団体、船舶、住所、識別番号、出生地、出生日、市民権、または国籍を示すエントリ ID。これは各政府機関から提供されます。
ListType	政府機関から提供されるリストの名前。SDN、EU、イングランド銀行、カナダ金融機関監督庁。
<b>リスク解析</b>	
OverAllRiskLevel	一致ごとのリスク ソース。0 ~ 16。詳細については、「 <a href="#">リスク解析スコアについて (430ページ)</a> 」を参照してください。
SanctionedCountryIdentified	認可国が一致すると認識されたかどうかを示します。値は Yes または No です。

### リスク解析スコアについて

これらの各入力に対しては、入力の有無と Global Sentry データベース内に一致するレコードがあるかどうかに応じて、リスク解析処理が一定のポイント値を割り当てます。リスク解析スコアは、これらのポイント値の合計です。ポイントがどのように割り当てられるかを次の表に示します。

表 9: リスク解析のスコアリング方法

入力	入力データなし	一致あり	一致なし
名前	0	4	0
住所	1	2	0
ID	1	2	0
誕生日	1	2	0
出生地	1	2	0
市民権	1	2	0
国籍	1	2	0

一般に、データベースと一致した各入力には 2 ポイントが割り当てられます。ただし、名前は例外です。名前の一致は 4 ポイントとカウントされます。名前スコアの加重が高いのは、OFAC などの情報源から得られる指針に従っているからで、そこでは名前の一致はその他の種類の一致よりもより重要であるとされています。

一方、入力があっても、データベースのどのエントリとも一致しない場合、割り当てられるポイントは 0 で、総合リスク水準には影響しません。これは、名前の一致があっても、その他の相当量のデータがデータベース内のエントリと一致しなければ、その一致を特定リストとの照合による "ヒット" とは見なすべきでないという指針に従うものです。

データの無い入力に 1 ポイントが設定されています。これには、1 つ以上の入力がデータベースと一致する一方で、マッチングに使用できる入力が一部欠けているようなトランザクションを、

よりリスクが高いトランザクションであると見なす効果があります。このタイプのトランザクションについては、データが欠けているが故に真のリスク水準を正確に計算できません。OFACなどの機関から得られる指針によれば、こうしたケースではトランザクションに関するリスクのより正確な評価を与えるために、欠けているデータの入手に努力することになっています。

スコアが高ければそれだけトランザクションのリスクも大きくなりますが、リスク水準だけで必ずしも適切なアクションを決定できるわけではありません。これはスコアが同じでも、一致あり、一致なし、データなしの入力について、さまざまな組み合わせが考えられるからです。禁止措置の妥当性を判断する補足的な情報を提供するために、Global Sentry サービスはマッチングで使われる7つの入力のそれぞれについて、さらに2つの指標を返します。これらは入力データの有無と、データベースの一致の有無を示します。これで、リスク範囲の中程にあるトランザクションに対して追加的な解析を行うことにより、ブラックリストの管理当局にトランザクションを報告する、正確なリスク評価のために追加的な入力データが必要とのマークをトランザクションに付ける、トランザクションを承認する、または、その他の何らかのアクションを行うなど、どの対処が妥当かを判断できます。

### Global Sentry サービスのカスタマイズ

Global Sentry は、Enterprise Designer で部分的に修正できる5つのデータフロー テンプレートを展開しています。個々のデータフローは、Spectrum™ Technology Platform、Universal Name、Data Normalization、Advanced Matching の各モジュールからインストールされた各種のコンポーネントによって構成されます。

以下のデータフローが使用可能です。

- 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

## Information Extraction モジュール

### InformationExtractor

InformationExtractor は、構造化されていないデータの文字列 (プレーンテキスト) から、名前や住所などのエンティティを抽出します。

入力のタイプによって精度が異なるため、選択されたタイプに対して一部のエンティティは返されない可能性があります。Information Extractor は自然言語処理を使用するため、ニュース記事やブログからの文法的に正しい文を含む文字列は、名前や日付の単純なリストよりも、高い精度で名前が返される可能性があります。

### リソース URL

JSON エンドポイント:

```
http://server:port/rest/InformationExtractor/result.json
```

XML エンドポイント:

```
http://server:port/rest/InformationExtractor/result.xml
```

### JSON 応答の例

この例では、JSON 応答を要求します。

```
http://myserver:8080/rest/InformationExtractor/result.json?
Data.PlainText=My+name+is+Arthur+Pitney&Option.EntityList=Person
```

この要求から次のような JSON が返されます。

```
{ "output_port": [ {
  "Entity": [ {
    "Text": "Aurthur Pitney",
    "Type": "Person"
  } ],
  "user_fields": []
} ] }
```

## XML 応答の例

この例では、XML 応答を要求します。

```
http://myserver:8080/rest/InformationExtractor/result.xml?
Data.PlainText=My+name+is+Arthur+Pitney&Option.EntityList=Person
```

この要求から次のような XML が返されます。

```
<xml.InformationExtractorResponse
xmlns="http://www.pb.com/spectrum/services/InformationExtractor">
  <output_port>
    <Result>
      <Entity>
        <Entity>
          <Text>Aurthur Pitney</Text>
          <Type>Person</Type>
        </Entity>
      </Entity>
      <user_fields/>
    </Result>
  </output_port>
</xml.InformationExtractorResponse>
```

## 要求

### 入力データのパラメータ

InformationExtractor は、構造化されていないデータを入力として受け取ります。

表 10 : 入力フォーマット

パラメータ	説明
Data.PlainText	情報の抽出元となるデータを表す構造化されていない文字列です。

### オプション

InformationExtractor ステージを使用すると、出力データのためのエンティティを選択できます。このステージに取り込まれたエンティティ タイプについては、属性が自動的に割り当てられます。ただし、クイック追加機能を使用して次の 15 の属性の任意のものまたはすべてを選択できます。

パラメータ	説明
Option.CategorizerName	テキスト分類に使用するモデルを指定します。
Option.CategoryCount	出力するカテゴリのマッチングレベル数(最も近い一致、最も近いものと2番目に近いもの、など)を指定します。
Option.EntityList	<p>構造化されていない文字列から抽出するデータのタイプを指定します。</p> <p>1つまたは複数指定します。各エンティティタイプの区切りにはカンマを使用します。</p> <ul style="list-style-type: none"><li><b>Address</b></li><li><b>CreditCard</b></li><li><b>Date</b></li><li><b>Email</b></li><li><b>HashTag</b></li><li><b>ISBN</b></li><li><b>Location</b></li><li><b>Mention</b></li><li><b>Organization</b></li><li><b>Person</b></li><li><b>Phone</b></li><li><b>ProperNouns</b></li><li><b>SSN</b></li><li><b>WebAddress</b></li><li><b>ZipCode</b></li></ul>
Option.OutputEntityCount	<p>特定のエンティティが現れた回数を出力で返すかどうかを指定します。</p> <ul style="list-style-type: none"><li><b>true</b> 構造化されていない文字列内に見つかったエンティティの数を返します。</li><li><b>false</b> 構造化されていない文字列内に見つかったエンティティの数を返しません。</li></ul>



## 応答

InformationExtractor からの出力は、入力文字列内に見つかったエンティティの一覧です。例えば、"Person" というエンティティ タイプを選択した場合、出力は入力文字列内に見つかった名前の一覧になります。同様に、"Date" というエンティティ タイプを選択した場合は、入力文字列内に見つかった日付の一覧が出力になります。それぞれのエンティティ (名前、住所、日付など) は、たとえ入力文字列内に複数回現れていても、一度しか返されません。

応答要素	説明
Text	文字列から抽出されたテキストです。
Type	抽出されたテキストのエンティティ タイプです。次のいずれかです。 <b>Address</b> <b>CreditCard</b> <b>Date</b> <b>Email</b> <b>HashTag</b> <b>ISBN</b> <b>Location</b> <b>Mention</b> <b>Organization</b> <b>Person</b> <b>Phone</b> <b>ProperNouns</b> <b>SSN</b> <b>WebAddress</b> <b>ZipCode</b>
Count	回数を返すオプションが有効になっている場合は、このフィールドに、その特定のエンティティが入力に現れた回数が含まれます。例えば、名前エンティティを返すことを選択していて、入力テキストに "John" という名前のインスタンスが 5 つ含まれている場合、出力には "John" が 1 回だけ含まれ、エンティティ タイプは "Name" に、出力回数は "5" になります。
Category	カテゴリーを使用した場合は、入力ファイル内の各レコードに対する予測カテゴリーです。

応答要素	説明
Rank	カテゴリザを使用した場合は、最多カウントから最少カウントまでのカテゴリのランクです。

## Spatial モジュール

### ドキュメントの場所

Location Intelligence モジュールは、任意の位置、対象領域、または対象地点やその他のビジネスデータ間の関係を決定し、そうした関係をマップ上に視覚的に示すことができる空間サービスを提供します。これらのサービスには次のものが含まれます。

- Geometry
- Feature
- Mapping
- MapTiling
- Named Resource
- Web Feature Service
- Web Map Service

Location Intelligence モジュール サービスの詳細については、『*Spectrum Spatial ガイド*』([support.pb.com](http://support.pb.com) にあります)を参照してください。

### ルーティング デモ ページとは

ルーティング デモ ページは、Spatial モジュールのルーティング REST サービスを用いた基本的なルーティング機能を実践的に試すためのインタラクティブなユーザ インターフェイスです。REST リクエストをサービス エンドポイントに送信し、使用されるリクエストを表示し、マップ上にレスポンスを示します。

現時点では、GetTravelBoundary と GetRoute の 2 つの REST サービスがルーティング デモ ページで取り上げられています。

### GetTravelBoundary

GetTravelBoundary は、ロケーションから一定の時間または距離で、車両または徒歩で到達可能なエリアの境界を示します。この機能では、時間到達圏または距離到達圏の計算に基づくポリゴンを取得します。時間到達圏とは、始点から一定時間内に道路を通過して到達できるエリアを表すポリゴンまたはポイントの集まりです。距離到達圏とは、始点から一定距離の移動で到達できる

エリアを表すポリゴンまたはポイントの集まりです。GetTravelBoundary 操作 (到達圏検索とも呼ばれます) は、始点、単位 (距離または時間)、および 1 つ以上のコストを入力として受け取り、生成された到達圏 (ポリゴン) を返します。コストとは、到達圏計算に使用する時間または距離の値を意味します。また、複数のコストを入力として提供することもできます。複数のコストの場合、コストはカンマ区切り文字列として提供することもできます。GetTravelBoundary REST の全入力パラメータの詳しい説明については、[GetTravelBoundary](#) (149ページ) を参照してください。

### GetRoute

GetRoute は、2 つのポイントまたは複数のポイントに対するルーティング情報を返します。始点および終点と、オプションで通過点を入力として取り、最短時間または最短距離となるルートを返します。GetRoute REST の全入力パラメータの詳しい説明については、[GetRoute](#) (108ページ) を参照してください。

#### ルーティング デモ ページの起動

次の手順で、ルーティング デモ ページを Web ブラウザから起動します。

1. ようこそページの [Spectrum Spatial] セクションから [サンプル アプリケーション] タブに移動し、[ルーティング デモ ページ] の下の [ルーティング デモ ページを開く] をクリックします。または、ルーティング デモ ページ Web アプリケーションの URL (<http://<サーバー名>:<ポート>/routingdemopage/>) をブラウザのアドレス バーに入力します。
2. 自分のユーザ名とパスワードでサインインします。これらは Spectrum™ Technology Platform で使用するのと同じ資格情報です。
3. [次へ] をクリックします。

注：間違った資格情報を入力すると、サインイン ページに戻り、ユーザ名またはパスワードが無効であることを示すエラー メッセージが表示されます。

ルーティング デモ ページが表示されます。

#### ルーティング デモ ページの使用

ルーティング機能を実践的に試すためのインタラクティブなインターフェイスを使用する際には、以下の手順に従います。

- **[データベース]**- デモで使用するルーティングデータベースリソースを変更します。データベースリソースを変更するには、ドロップダウンリストからデータベースリソースを選択します。Spectrum で設定したリソースのみが、このリストに表示されます。
- **サービス タブ**- デモするサービスとして、Travel Boundary または Route を選択します。
- **[ポイント]**- どちらのサービスも、操作でポイントを使用します (基点、始点、終点)。ポイント情報を手動で入力するか、マップ上の位置を右クリックしてポイントの種類を選択します。
- **[詳細オプション]**- その他のルーティングオプションをリクエストに追加します。その他のルーティング オプションを追加するには、**[詳細オプション]** をクリックしてオプションを指定しま

す。サービス REST パラメータとオプションの詳細については、[GetTravelBoundary](#) (149ページ) または [GetRoute](#) (108ページ) を参照してください。

- **[適用]** - REST リクエストを開始して、マップイメージを返し、マップの生成に使用した REST リクエストを表示します。リクエストを行うには、すべての必須パラメータ (\*) を指定して **[適用]** をクリックします。
- **[クリア]** - マップイメージ、フィールド データ、またはその両方を削除します。マップまたはフィールド データ上のルーティング情報をクリアするには、**[クリア]** ドロップダウンリストをクリックして、**[すべて]**、**[データ]**、または **[マップ]** を選択します。
- **[ユーザの変更]** - 現在のログイン ユーザを変更します。ユーザを変更するには、**[ユーザの変更]** ボタンをクリックしてユーザ名とパスワードを入力し、**[次へ]** をクリックします。

## GetRoute

### 説明

GetRoute サービスは、2つのポイントまたは複数のポイントに対するルーティング情報を返します。始点および終点とオプションで通過点を入力として取り、最短時間または最短距離となるルートを返します。

注：REST サービスからの応答は JSON 形式で、返されるジオメトリは GeoJSON 形式です。要求の GET URL に無効なクエリ パラメータが含まれていたり、POST のペイロードが無効である場合は、累積的なエラー応答が JSON 配列に返されます。応答 JSON の value ノードは非推奨です。エラーの確認には、errors ノードを使用してください。

### HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。HTTP GET は、その他の JSON ペイロードを必要としない単純なルートに使用します。HTTP GET 要求に通過点を追加することもできます。

```
HTTP GET
/rest/Spatial/erm/databases/dbsource.json?q=route&query_parameters
```

ここで、**dbsource** は、ルート用のデータを含むデータベースの名前です。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### HTTP POST URL 形式

HTTP POST リクエストには次の形式を使用します。

```
HTTP POST:
/rest/Spatial/erm/databases/dbsource.json?q=route&query_parameters
```

```
POST BODY: Content-Type:application/json {Route Data}
```

Route Data は、通過点、一時更新、または道路タイプの優先度を含む計算で使用されるその他のルート情報を表す POST Json 本体 (Content-Type: application/json) です。これらのオプションの詳細については、[GetRoute HTTP POST オプション \(118ページ\)](#) を参照してください。

### クエリ パラメータ

この演算は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
startPoint	文字列	はい	ルートの始点。形式は x,y,coordSys です。例: -74.2,40.8,EPSG:4326
endPoint	文字列	はい	ルートの終点。形式は x,y,coordSys です。例: -74.2,40.8,EPSG:4326
intermediatePoints	文字列	いいえ	ルートに沿って含める通過点のリスト。これを HTTP GET 要求に含めるには、形式 Long,Lat,Long,Lat,...,coordsys を使用します。例えば、-74.2,40.8,-73,42,EPSG:4326 を使用します。一連の通過点を HTTP POST 要求に含めるには、ルートのポイントを示す MultiPoint JSON ペイロードを追加します。通過点が URL と JSON ペイロードの両方で指定されている場合は、JSON ペイロードが優先され、URL の通過点は無視されます。
oip	Boolean	いいえ	通過点を最適化する必要があるかどうかを示す処理パラメータ。デフォルトは false です。デフォルトでは、通過点は計算で指定された順序で使用されます。true に設定すると、指定されたポイントはルートの計算時に最適な順序に変更されます。
returnIntermediatePoints	Boolean	いいえ	ルートの応答で通過点を返すかどうか。デフォルトは false です。  true または false 以外の値を指定した場合は、デフォルトの false に設定されます。このオプションは、POST 本体で指定された順序で通過点を返します。オプション oip の値が true に設定されている場合、このオプションは最適化された順序で通過点を返します。

パラメータ	タイプ	必須	説明
destinationSrs	文字列	いいえ	ルートと結果のジオメトリを返す座標系。既定値は使用するデータの座標系です。
optimizeBy	文字列	いいえ	ルートに使用する最適化のタイプ。有効な値は <i>time</i> または <i>distance</i> です。既定値は <i>time</i> です。
returnDistance	Boolean	いいえ	道順に移動距離を含めます。デフォルトは <b>true</b> です。
distanceUnit	文字列	いいえ	距離を返すときの単位。既定値は <i>m</i> (メートル) です。使用可能な値: <i>m</i> (メートル)、 <i>km</i> (キロメートル)、 <i>yd</i> (ヤード)、 <i>ft</i> (フィート)、 <i>mi</i> (マイル)。
returnTime	Boolean	いいえ	道順に、その道順に従った場合にかかる時間を含めます。デフォルトは <b>true</b> です。
時間単位	文字列	いいえ	時間を返すときの単位。既定値は <i>min</i> (分) です。使用可能な値: <i>min</i> (分)、 <i>msec</i> (ミリ秒)、 <i>s</i> (秒)、 <i>h</i> (時間)。

パラメータ	タイプ	必須	説明
language	文字列	いいえ	

パラメータ	タイプ	必須	説明
			道順を返すときの言語。道順を返す場合 (directionsStyle が Normal または Terse に定義されている場合) にのみ使用します。既定値は英語 (en) です。
			道順は次の言語で返すことができます。
			<b>sq</b> アルバニア語で道順を返します。
			<b>zh_CN</b> 中国語で道順を返します。
			<b>zh_TW</b> 中国語 (台湾) で道順を返します。
			<b>hr</b> クロアチア語で道順を返します。
			<b>cs</b> チェコ語で道順を返します。
			<b>da</b> デンマーク語で道順を返します。
			<b>nl</b> オランダ語で道順を返します。
			<b>en</b> 英語で道順を返します。デフォルト
			<b>en-US</b> アメリカ英語で道順を返します。
			<b>et</b> エストニア語で道順を返します。
			<b>fi</b> フィンランド語で道順を返します。
			<b>fr</b> フランス語で道順を返します。
			<b>de</b> ドイツ語で道順を返します。
			<b>hu</b> ハンガリー語で道順を返します。
			<b>it</b> イタリア語で道順を返します。
			<b>ja</b> 日本語で道順を返します。
			<b>lv</b> ラトビア語で道順を返します。
			<b>lt</b> リトアニア語で道順を返します。
			<b>no</b> ノルウェー語で道順を返します。
			<b>pt</b> ポルトガル語で道順を返します。
			<b>ro</b> ルーマニア語で道順を返します。
			<b>sk</b> スロバキア語で道順を返します。
			<b>sl</b> スロベニア語で道順を返します。
			<b>es</b> スペイン語で道順を返します。
			<b>sv</b> スウェーデン語で道順を返します。
			<b>ru</b> ロシア語で道順を返します。



パラメータ	タイプ	必須	説明
			<b>tr</b> トルコ語で道順を返します。
returnDirectionGeometry	Boolean	いいえ	ルートの応答で、ルート指示に関連付けられたジオメトリを返します。デフォルトは <b>false</b> です。
directionsStyle	文字列	いいえ	返す道順のタイプ。デフォルトは <b>None</b> です。道順を返す必要がある場合は、このパラメータを指定します。道順を指定するときのオプションは次のとおりです。  <b>None</b> 道順を返しません。指定されない場合の既定値です。  <b>Normal</b> 道順は Web ベース アプリケーションに適した完全な形式で返されます。  <b>Terse</b> 道順はモバイル アプリケーションに適した短縮形式で返されます。
segmentGeometryStyle	文字列	いいえ	ルートのセグメントを表すジオメトリの形式。デフォルトは <b>None</b> です。セグメント ジオメトリを返す必要がある場合は、このパラメータを指定します。道順を指定するときのオプションは次のとおりです。  <b>None</b> セグメントのジオメトリ表現を返します。指定されない場合の既定値です。  <b>End</b> ルートの各セグメントと終点を <b>LineString</b> で返します。  <b>All</b> 各セグメントとすべての形状ポイントを <b>LineString</b> で返します。 <b>LineString</b> はマップ上でオーバーレイとして使用できます。
primaryNameOnly	Boolean	いいえ	道順の中の任意の通りのすべての名前を返すか、通りの主要名のみを返すかを指定します。道順を返す場合にのみ使用します。既定値は <b>false</b> です。
majorRoads	Boolean	いいえ	計算時に、すべての道路を含めるか、幹線道路のみを含めるかを指定します。幹線道路のみを含めるように指定した場合は、パフォーマンスは向上しますが精度は低くなる場合があります。デフォルトは <b>false</b> です。

パラメータ	タイプ	必須	説明
historicTrafficTimeBucket	文字列	いいえ	<p>ルーティング計算において、旅行速度履歴を使用するかどうかを指定します。これらの速度は、1日のうちの異なる時間帯に基づきます。この機能を使用するには、旅行速度履歴がデータに含まれている必要があります。各国/地域のデータに対する時間帯定義は同じです。時間帯に対する速度は異なる場合があります。次のオプションがあります。</p> <p><b>None</b> デフォルト値です。計算において、旅行速度履歴は使用しません。代わりに速度の平均値を使用します。</p> <p><b>AMPeak</b> 午前のピーク速度でルートを計算します。午前ピークの時間帯は、1日のうちの 07:00 ~ 10:00 です。</p> <p><b>PMPeak</b> 午後のピーク速度でルートを計算します。午後ピークの時間帯は、1日のうちの 16:00 ~ 19:00 です。</p> <p><b>OffPeak</b> オフピーク (日中) 速度でルートを計算します。オフピークの時間帯は、1日のうちの 10:00 ~ 16:00 です。</p> <p><b>Night</b> 夜間速度でルートを計算します。夜間の時間帯は、1日のうちの 22:00 ~ 04:00 です。</p>
除外	文字列	いいえ	<p>ルート計算中に回避する道路タイプのリストをカンマで区切って指定します。これは文字列パラメータです。パラメータの値として道路タイプを指定すると、ルート計算でルートからそのタイプの道路が除外されます。例えば、パラメータの値として<b>有料道路</b>を指定すると、計算されたルートから有料道路が除外されます。</p>
バージョン	文字列	いいえ	<p>GetRoute REST サービスのバージョンを指定します。有効な値は 1 と 2 です。version のデフォルト値は 1 です。</p>

パラメータ	タイプ	必須	説明
localRoadsLoadFactor	文字列	いいえ	<p>ルート計算またはマトリクス計算中にメモリにロードできる地方道路の数を指定します。ロードできる道路の数は、このパラメータで選択した値に正比例します。パラメータの最小値は 1 で、最大値は 3 です。有効な値は 1、2、3 です。デフォルトは 1 です。パラメータがルート計算またはマトリクス計算に与える影響の詳細については「<a href="#">地方道路の負荷係数</a>」を参照してください。</p> <p>注：パラメータに小数の値を指定することはできません。</p>

### 例

始点と終点を持つ単純なルート。

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?route=startPoint=-73.97,40.79,eps:4326&endPoint=-73.98,40.74,eps:4326
```

応答

```
{
  "distance": 7779,
  "distanceUnit": "m",
  "time": 16.75,
  "timeUnit": "min"
}
```

通過点を持つルート。

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?q=route&
startPoint=-73.970257,40.794045,eps:4326&endPoint=-
73.972103,40.786605,eps:4326&intermediatePoints=-73.976266,40.788717,
-73.973562,40.792193,-73.971802,40.794630,eps:4326&oip=true&
returnIntermediatePoints=true"/>
```

応答

```
{
```

```

    "distance": 1921,
    "distanceUnit": "m",
    "intermediatePoints": {
      "type": "MultiPoint",
      "coordinates": [
        [-73.971802, 40.79463],
        [-73.973562, 40.792193],
        [-73.976266, 40.788717]
      ]
    },
    "time": 4.2,
    "timeUnit": "min"
  }

```

道順が有効になっているルート。

```

http://www.p.com/rest/Spatial/en/databases/usroutebase.json?routeStartPoint=-73.97,40.79,eps:4326&endPoint=-73.98,40.74,eps:4326&language=en&directionsStyle=Normal&returnDirectionGeometry=true

```

応答

```

{
  "time": 10.58,
  "timeUnit": "min",
  "distance": 9035,
  "distanceUnit": "m",
  "language": "en",
  "directionsStyle": "Normal",
  "routeDirections": [
    {
      "time": 0.03,
      "timeUnit": "min",
      "distance": 25,
      "distanceUnit": "m",
      "instruction": "",
      "directionGeometry": {
        "type": "LineString",
        "coordinates": [
          [
            [-76.421169,
              42.69302
            ],
            [
              [-76.421353,
                42.692645
              ],
              ...
            ]
          ]
        ]
      }
    }
  ]
}

```

```

},
{
  "time": 0.7,
  "timeUnit": "min",
  "distance": 394,
  "distanceUnit": "m",
  "instruction": "Turn right on W 91st St and travel West 394.0 m (0.7 min).",
  "directionGeometry":
  {
    "type": "LineString",
    "coordinates":
    [
      [
        -76.429896,
        42.67153
      ],
      ...
    ]
  }
}
]
}
}
}
}

```

### バージョン固有のエラー応答

リクエストに無効なパラメータ値 (ポイントが境界の外側にある、など) を入力すると、入力したバージョンによって異なるエラー応答が返ります。バージョンが 1 の場合、値とエラーが返りますが、バージョンが 2 の場合、応答にはエラーのみが含まれます。

- version が 1 の場合のリクエスト:

```

http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
&q=route&startPoint=-14.321600,60.662859,epsg:4326&endPoint=-74.035208,40.695624,
epsg:4326&distanceUnit=km&version=1

```

- レスポンス:

```

{
  "value": "Point outside boundaries: (-14.3216,60.662859,0)",
  "errors": [
    {
      "errorCode": 5008,
      "userMessage": "Point outside boundaries: (-14.3216,60.662859,0)"
    }
  ]
}

```

- version が 2 の場合のリクエスト:

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
&q=route&startPoint=-14.321600,60.662859,epsg:4326&endPoint=-74.035208,40.695624,
epsg:4326&distanceUnit=km&version=2
```

- レスポンス:

```
{
  "errors": [
    {
      "errorCode": 5008,
      "userMessage": "Point outside boundaries: (-14.3216,60.662859,0)"
    }
  ]
}
```

### GetRoute HTTP POST オプション

#### HTTP POST URL 形式

標準の HTTP GET パラメータのほか、通過点、一時更新、および道路タイプの優先度を指定する HTTP POST ペイロード オプションをリクエストに追加できます。コンテンツ タイプは application/json に設定する必要があります。HTTP POST リクエストには次の形式を使用します。

```
HTTP POST:
/rest/Spatial/erm/databases/dbsource.json?q=route&query_parameters
POST BODY: Content-Type:application/json {Route Data}
```

#### 通過点

ルートに沿って含める通過点のリスト。一連の通過点を HTTP POST 要求に含めるには、ルートのポイントを示す MultiPoint JSON ペイロードを追加します。通過点が URL と JSON ペイロードの両方で指定されている場合は、JSON ペイロードが優先され、URL の通過点は無視されます。

例: 通過点の HTTP POST ペイロード

```
{
  "intermediatePoints": {"type": "MultiPoint","crs": {"type":
"name","properties": {"name": "epsg:4326"}}, "coordinates": [[
-73.976266,40.788717],[ -73.973562,40.792193],[ -73.971802,40.794630]]}
}
```

## 一時更新

この優先設定セットを使用して、要求ごとに一時更新を設定することができます。例えば、サーバーがすべての幹線道路タイプを回避するよう要求することができます。要求ごとに1つ以上の更新を含めることができます。速度更新の場合、正の速度値は速度の増加、負の速度値は速度の減少です。次は、一時更新タイプの説明です。

更新タイプ	説明
point	ポイントの更新は、対応するポイント (緯度、経度) に適用される変更です。特定のポイントに対して、ポイントの除外、ポイントの速度の設定、値または割合によるポイントの速度の変更 (増加または減少) を行うことができます。次のいずれかの更新タイプを使用します。
<b>percentage</b>	速度を増加 (正の値) または減少 (負の値) させる割合を指定してポイントの速度の増加を定義する速度更新です。
<b>speed</b>	速度単位と新しい速度を指定してポイントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には kph (キロメートル/時)、mph (マイル/時)、mps (メートル/秒)、mtpm (メートル/分) のいずれかの値を指定できます。
<b>speedAdjustment</b>	速度 (単位と値) の変更を指定してポイントの速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。速度更新の場合、速度単位には kph (キロメートル/時)、mph (マイル/時)、mps (メートル/秒)、mtpm (メートル/分) のいずれかの値を指定できます。
<b>exclude</b>	指定されたポイントをルート計算から除外する文字列値です。ポイントを除外するには、ポイントを指定して、Y と定義した <b>exclude</b> パラメータを含める必要があります。有効な値は Y (はい) と N (いいえ) です。

更新タイプ

説明

---

segmentID



## 更新タイプ

## 説明

セグメントの更新は、対応するセグメント ID に適用される変更です。特定のセグメントに対して、セグメントの除外、セグメントの速度の設定、値または割合によるセグメントの速度の変更 (増加または減少)、セグメントの道路タイプの変更を行うことができます。次のいずれかの更新タイプを使用します。

**percentage** 速度を増加 (正の値) または減少 (負の値) させる割合を指定して `segmentID` の速度の増加を定義する速度更新です。

**speed** 速度単位と新しい速度を指定して `segmentID` の新しい速度を定義する速度更新です。速度更新の場合、速度単位には `kph` (キロメートル/時)、`mph` (マイル/時)、`mps` (メートル/秒)、`mtpm` (メートル/分) のいずれかの値を指定できます。

**speedAdjustment** 速度 (単位と値) の変更を指定して `segmentID` の速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。速度更新の場合、速度単位には `kph` (キロメートル/時)、`mph` (マイル/時)、`mps` (メートル/秒)、`mtpm` (メートル/分) のいずれかの値を指定できます。

**exclude** 指定された `segmentID` をルート計算から除外する文字列値です。`segmentID` を除外するには、`segmentID` を指定して、`Y` と定義した `exclude` パラメータを含める必要があります。有効な値は `Y` (はい) と `N` (いいえ) です。

**roadType** ルート計算に使用するセグメントの道路タイプの値を変更する文字列値です。

`roadType` には、次の種類があります。

- 進入路
- 裏道路
- 接続道路
- フェリー
- 小道
- 侵入制限道路 (密集都市部)
- 侵入制限道路 (農村部)
- 侵入制限道路 (郊外)
- 侵入制限道路 (都市部)
- 地方道路 (密集都市部)
- 地方道路 (農村部)
- 地方道路 (郊外)
- 地方道路 (都市部)
- 主要地方道路 (密集都市部)
- 主要地方道路 (農村部)
- 主要地方道路 (郊外)
- 主要地方道路 (都市部)

## 更新タイプ

## 説明

- 
- 幹線道路 (密集都市部)
  - 幹線道路 (農村部)
  - 幹線道路 (郊外)
  - 幹線道路 (都市部)
  - 補助地方道路 (密集都市部)
  - 補助地方道路 (農村部)
  - 補助地方道路 (郊外)
  - 補助地方道路 (都市部)
  - 一般道路 (密集都市部)
  - 一般道路 (農村部)
  - 一般道路 (農村部)
  - 一般道路 (都市部)
  - 主要高速道路 (密集都市部)
  - 主要高速道路 (農村部)
  - 主要高速道路 (郊外)
  - 主要高速道路 (都市部)
  - 出入路 (密集都市部)
  - 出入路 (侵入制限道路)
  - 出入路 (幹線道路)
  - 出入路 (主要高速道路)
  - 出入路 (農村部)
  - 出入路 (一般高速道路)
  - 出入路 (都市部)
  - 出入路 (郊外)
  - 一般高速道路 (密集都市部)
  - 一般高速道路 (農村部)
  - 一般高速道路 (郊外)
  - 一般高速道路 (都市部)
-

## 更新タイプ

## 説明

## 道路タイプ

道路タイプの更新は、対応する道路タイプに適用される変更です。特定の道路タイプに対して、道路タイプの速度の設定、値または割合による道路タイプの速度の変更 (増加または減少) を行うことができます。次のいずれかの更新タイプを使用します。

- percentage** 速度を増加 (正の値) または減少 (負の値) させる割合を指定して道路タイプの速度の増加を定義する速度更新です。
- speed** 速度単位と新しい速度を指定して道路タイプの新しい速度を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。
- speedAdjustment** 速度 (単位と値) の変更を指定して道路タイプの速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

## 例: 一時更新の HTTP POST ペイロード

```
{
  "transientUpdates": [
    {
      "segmentID": "7e3396fc:151186f",
      "updates": [
        { "percentage": 26.0 }
      ]
    },
    {
      "point": { "type": "Point",
        "crs": {
          "type": "name",
          "properties": {
            "name": "epsg:4326"
          }
        }
      },
      "coordinates": [
        -73.972776,
        40.795076
      ]
    }
  ]
}
```

```

    },
    "updates": [
      { "speedAdjustment" : { "velocity": 5, "velocityUnit": "kph"} }
    ]
  },
  {
    "roadType": "major road dense urban",
    "updates": [
      { "speed": { "velocity": 25, "velocityUnit": "kph"} }
    ]
  }
]
}

```

### 車両規制

車両規制は、ルートに沿って移動する特定車両の動作と属性をガイドするルーティングエンジンへの規則で構成されます。指定された車両の属性(高さ、幅、長さ、重量など)、および道路網に見られる車両規制の属性に応じて、特定の車両に、あるセグメント上を経由させることができるかどうか決定されます。道路網に車両規制の属性がない場合、入力制限パラメータは、結果的なルートに影響を与えません。

以下に示すのは、車両規制の一連のパラメータです。

#### オプション

#### 説明

##### looseningBarrierRestrictions

ルートを決定する際、通行規制を除去するかどうかを指定します。このような規制は、地方の条例によって特定車両がセグメントの通行を禁止されている場合や、必要なときのみ(ラストマイルアクセスや地方の配達など)特定車両がそのセグメントで許可されている場合がほとんどです。通行規制が除去されたルートは、通行規制のないルートより短い速い場合でも、引き続きルートコストが高くなります。

## オプション

## 説明

## vehicleAttributes

ルートを決定する際、タイプ、高さ、重量、長さ、または幅に基づいて規制される車両の詳細を指定します。特定車両は、ショートトレーラーから長い3連結トレーラーまでさまざまなタイプに分かれています。車両規制属性は、車両タイプごとにまとめられます。つまり、あるセグメントに対して1つの車両タイプを優先すると共に、同じセグメントで別の車両タイプを規制することが完全に可能です。次のタイプの車両情報を使用します。

- vehicleType** ALL または次のいずれかのタイプの車両を選択します:  
STRAIGHT、SEMI\_TRAILOR、STANDARD\_DOUBLE、INTERMEDIATE\_DOUBLE、LONG\_DOUBLE、TRIPLE、OTHER\_LONG\_COMBINATION\_VEHICLE。
- weight** 車両の最大重量を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。重量の単位は、kg、lb、mt、tです。
- height** 車両の最大高を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。高さの単位は、ft、yd、mi、m、kmです。
- length** 車両の最大長を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。長さの単位は、ft、yd、mi、m、kmです。
- width** 車両の最大幅を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。幅の単位は、ft、yd、mi、m、kmです。

注：重量/高さ、または長さ/幅のいずれかを、対応する単位と共に指定する必要があります。

## 例

## 車両規制なし

リクエスト:

```
HTTP GET
/rest/Spatial/erm/databases/US_CVR.json?q=routeCostMatrix&startPoints=-74.7221203,42.9737073,eps:4326&endPoints=-74.6671887,42.8097083,eps:4326
```

レスポンス:

```
{
  "matrix": [{
    "distance": 40025,
```

```

    "distanceUnit": "m",
    "endPoint": {
      "type": "Point",
      "coordinates": [-74.6671887, 42.8097083]
    },
    "startPoint": {
      "type": "Point",
      "coordinates": [-74.7221203, 42.9737073]
    },
    "time": 36.57,
    "timeUnit": "min"
  }
}

```

## 車両規制あり

リクエスト:

```

HTTP GET
/rest/Spatial/erm/databases/US_CVR.json?q=routeCostMatrix&startPoints=-74.7221203,42.9737073,eps:4326&
endPoints=-74.6671887,42.8097083,eps:4326

```

車両規制の HTTP POST ペイロード

```

{
  "cvr":{
    "looseningBarrierRestrictions":"n","vehicleAttributes":{
      "vehicleType":"ALL","heightUnit":"meter","height":"4","weightUnit":"Kilogram","weight":"40000"
    }
  }
}

```

レスポンス:

```

{
  "matrix": [{
    "distance": 44933,
    "distanceUnit": "m",
    "endPoint": {
      "type": "Point",
      "coordinates": [-74.6671887, 42.8097083]
    },
    "startPoint": {
      "type": "Point",
      "coordinates": [-74.7221203, 42.9737073]
    },
    "time": 37.48,
  }
}

```

```

    "timeUnit": "min"
  }]
}

```

CVR ありでは距離と時間の値が変化しています。

### 道路タイプの優先度

ルート決定時にさまざまな道路に与える優先順位を指定します。次は、道路タイプの優先度オプションの説明です。

オプション	説明
高	この道路タイプを他の道路タイプよりも優先します。
中	この道路タイプに他の道路タイプと同じ優先設定を与えます。道路タイプに優先設定を指定しないと、デフォルトとして "中" が使われます。
低	他の道路タイプをこの道路タイプよりも優先します。
除外	この道路タイプを可能な限りルートで使用しません。特定の道路タイプを道順から常に除外できるとは限りません。状況によっては、回避対象の道路タイプの代わりになるものが非常に貧弱なために回避対象の道路タイプを使うルートが選ばれることがあります。また、始点または終点がセグメントの横にあって、そのセグメントの道路タイプが回避されている場合も、そのセグメントが使われます。

### 例: 道路タイプの優先度の HTTP POST ペイロード

```

{
  " roadTypesPriority ": {
    "RoadType.MajorRoadDenseUrban": "High",
    "RoadType.LimitedAccessDenseUrban": "Low",
    "RoadType.LimitedAccessRural": "Medium",
    "RoadType.PrimaryHighwayUrban": "Avoid"
  }
}

```

```
}

```

## GetRouteCostMatrix

### 説明

GetRouteCostMatrix サービスは、始点と終点の間の移動時間と距離を計算し、最短時間または最短距離となるルート返します。この結果は、個々のルートの合計時間と合計距離 (ルート コスト) を示します。例えば、4 つの始点と 4 つの終点を入力すると、合計 16 個のルートが計算されます。

注： REST サービスからの応答は JSON 形式で、返されるジオメトリは GeoJSON 形式です。要求の GET URL に無効なクエリ パラメータが含まれていたり、POST のペイロードが無効である場合は、累積的なエラー応答が JSON 配列に返されます。応答 JSON の value ノードは非推奨です。エラーの確認には、errors ノードを使用してください。

### HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。HTTP GET は、その他の JSON ペイロードを必要としない単純なコスト計算に使用します。

```
HTTP GET
/rest/Spatial/erm/databases/dbsource.json?q=routeCostMatrix&query_parameters
```

ここで、**dbsource** は、ルート用のデータを含むデータベースの名前です。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### HTTP POST URL 形式

HTTP POST リクエストには次の形式を使用します。

```
HTTP POST:
/rest/Spatial/erm/databases/dbsource.json?q=routeCostMatrix&query_parameters
POST BODY: Content-Type:application/json {Route Data}
```

Route Data は、入力ポイントのリストが呼び出し元の URL バッファの制限を超える場合の計算や、一時更新、または道路タイプの優先度を含む計算で使用されるその他のルート情報を表す POST Json 本体 (Content-Type: application/json) です。これらのオプションの詳細については、[GetRouteCostMatrix HTTP POST オプション](#) (133ページ) を参照してください。

### クエリ パラメータ

この演算は、次のクエリ パラメータを取ります。



パラメータ	タイプ	必須	説明
始点	文字列	はい	ルートの始点。形式は <code>long,lat,long,lat,...,coordSys</code> です。例えば、 <code>-74.2,40.8,-73,42,EPSG:4326</code> を使用します。
終点	文字列	はい	ルートの終点。形式は <code>long,lat,long,lat,...,coordSys</code> です。例えば、 <code>-74.2,40.8,-73,42,EPSG:4326</code> を使用します。
<code>destinationSrs</code>	文字列	いいえ	ルートと結果のジオメトリを返す座標系。既定値は使用するデータの座標系です。
<code>optimizeBy</code>	文字列	いいえ	ルートに使用する最適化のタイプ。有効な値は <code>time</code> または <code>distance</code> です。既定値は <code>time</code> です。
<code>returnDistance</code>	Boolean	いいえ	道順に移動距離を含めます。デフォルトは <code>true</code> です。 <code>returnDistance</code> および <code>returnTime</code> パラメータの両方を同じ要求で <code>false</code> にはできません。
<code>distanceUnit</code>	文字列	いいえ	距離を返すときの単位。既定値は <code>m</code> (メートル) です。使用可能な値: <code>m</code> (メートル)、 <code>km</code> (キロメートル)、 <code>y</code> (ヤード)、 <code>ft</code> (フィート)、 <code>mi</code> (マイル)。
<code>returnTime</code>	Boolean	いいえ	道順に、その道順に従った場合にかかる時間を含めます。デフォルトは <code>true</code> です。 <code>returnDistance</code> および <code>returnTime</code> パラメータの両方を同じ要求で <code>false</code> にはできません。
時間単位	文字列	いいえ	時間を返すときの単位。既定値は <code>min</code> (分) です。使用可能な値: <code>min</code> (分)、 <code>msec</code> (ミリ秒)、 <code>s</code> (秒)、 <code>h</code> (時間)。
<code>majorRoads</code>	Boolean	いいえ	計算時に、すべての道路を含めるか、幹線道路のみを含めるかを指定します。幹線道路のみを含めるように指定した場合は、パフォーマンスは向上しますが精度は低くなる場合があります。デフォルトは <code>false</code> です。

パラメータ	タイプ	必須	説明
returnOptimalRoutesOnly	Boolean	いいえ	始点/終点の各組み合わせに対し、最適化ルートのみを返すかどうかを指定します。デフォルトは <b>true</b> です。最適化ルートは、 <b>optimizeBy</b> パラメータに応じて、最短時間ルートまたは最短距離のいずれかになります。
historicTrafficTimeBucket	文字列	いいえ	<p>ルーティング計算において、旅行速度履歴を使用するかどうかを指定します。これらの速度は、1日のうちの異なる時間帯に基づきます。この機能を使用するには、旅行速度履歴がデータに含まれている必要があります。各国/地域のデータに対する時間帯定義は同じです。時間帯に対する速度は異なる場合があります。次のオプションがあります。</p> <p><b>None</b> デフォルト値です。計算において、旅行速度履歴は使用しません。代わりに速度の平均値を使用します。</p> <p><b>AMPeak</b> 午前のピーク速度でルートを計算します。午前ピークの時間帯は、1日のうちの 07:00 ~ 10:00 です。</p> <p><b>PMPeak</b> 午後のピーク速度でルートを計算します。午後ピークの時間帯は、1日のうちの 16:00 ~ 19:00 です。</p> <p><b>OffPeak</b> オフピーク (日中) 速度でルートを計算します。オフピークの時間帯は、1日のうちの 10:00 ~ 16:00 です。</p> <p><b>Night</b> 夜間速度でルートを計算します。夜間の時間帯は、1日のうちの 22:00 ~ 04:00 です。</p>
除外	文字列	いいえ	<p>ルート計算中に回避する道路タイプのリストをカンマで区切って指定します。これは文字列パラメータです。パラメータの値として道路タイプを指定すると、ルート計算でルートからそのタイプの道路が除外されます。例えば、パラメータの値として有料道路を指定すると、計算されたルートから有料道路が除外されます。</p>

パラメータ	タイプ	必須	説明
バージョン	文字列	いいえ	GetRouteCostMatrix REST サービスのバージョンを指定します。有効な値は 1 と 2 です。version のデフォルト値は 1 です。
localRoadsLoadFactor	文字列	いいえ	<p>ルート計算またはマトリクス計算中にメモリにロードできる地方道路の数を指定します。ロードできる道路の数は、このパラメータで選択した値に正比例します。パラメータの最小値は 1 で、最大値は 3 です。有効な値は 1、2、3 です。デフォルトは 1 です。パラメータがルート計算またはマトリクス計算に与える影響の詳細については「<a href="#">地方道路の負荷係数</a>」を参照してください。</p> <p>注：パラメータに小数の値を指定することはできません。</p>

### 例

2つの始点と2つの終点を持つコストマトリクスルート。

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
endPoint=-73.56567672202618,40.554384822358614&
startPoint=-73.34345711862802,40.66688488742393&
localRoadsLoadFactor=2
```

### 応答

```
{
  "matrix": [{
    "distance": 35.258,
    "distanceUnit": "km",
    "endPoint": {
      "type": "Point",
      "coordinates": [-73.34345711862802, 40.66688488742393],
      "crs": {
        "type": "name",
        "properties": {
          "name": "epsg:4322"
        }
      }
    },
    "startPoint": {
      "type": "Point",
      "coordinates": [-73.56567672202618, 40.554384822358614],
      "crs": {
        "type": "name",
        "properties": {
```

```

    "name": "epsg:4322"
  }
}
}, {
  "distance": 41.761,
  "distanceUnit": "km",
  "endPoint": {
    "type": "Point",
    "coordinates": [-73.34345711862802, 40.66688488742393],
    "crs": {
      "type": "name",
      "properties": {
        "name": "epsg:4322"
      }
    }
  },
  "startPoint": {
    "type": "Point",
    "coordinates": [-73.46567684021008, 40.454384834155185],
    "crs": {
      "type": "name",
      "properties": {
        "name": "epsg:4322"
      }
    }
  }
}
]]
}

```

### バージョン固有のエラー応答

リクエストに無効なパラメータ値 (ポイントが境界の外側にある、など) を入力すると、入力したバージョンによって異なるエラー応答が返ります。バージョンが 1 の場合、値とエラーが返りますが、バージョンが 2 の場合、応答にはエラーのみが含まれます。

- version が 1 の場合のリクエスト:

```

http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
q=routeCostMatrix&startPoints=-73.56565,40.5545,-73.46565,40.4545,epsg:4326&
endPoints=-14.321600,60.662859,-73.14343,40.267,epsg:4326&version=1

```

- レスポンス:

```

{
  "value": "Point outside boundaries: (-14.3216,60.662859,0)",
  "errors": [
    {
      "errorCode": 5008,
      "userMessage": "Point outside boundaries: (-14.3216,60.662859,0)"
    }
  ]
}

```

```

    }
  ]
}

```

- version が 2 の場合のリクエスト:

```

http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
q=routeCostMatrix&startPoints=-73.56565,40.5545,-73.46565,40.4545,epsg:4326&
endPoints=-14.321600,60.662859,-73.14343,40.267,epsg:4326&version=2

```

- レスponse:

```

{
  "errors": [
    {
      "errorCode": 5008,
      "userMessage": "Point outside boundaries: (-14.3216,60.662859,0)"
    }
  ]
}

```

### GetRouteCostMatrix HTTP POST オプション

#### HTTP POST URL 形式

標準の HTTP GET パラメータのほか、一時更新や道路タイプの優先度を指定する HTTP POST ペイロード オプションをリクエストに追加できます。入力ポイントのリストが呼び出し元の URL バッファの制限を超える場合は、HTTP POST ペイロードを使用することもできます。コンテンツタイプは `application/json` に設定する必要があります。HTTP POST リクエストには次の形式を使用します。

```

HTTP POST:
/rest/Spatial/erm/databases/dbsource.json?q=routeCostMatrix&query_parameters
POST BODY: Content-Type:application/json {Route Data}

```

#### 始点と終点の定義

一連の始点または終点を HTTP POST 要求に含めるには、ルートのポイントを示す `MultiPoint JSON` ペイロードを追加します。始点と終点が HTTP POST ペイロードに定義されている場合、`startPoints` および `endPoints` パラメータは URL の必須クエリ パラメータではありません。これらのパラメータが URL に定義されていても無視されます。URL にあるポイントが無視されると、`spectrum-server.log` ファイルに警告メッセージが記録されます。

## 例: 始点の HTTP POST ペイロード

```
{
  "startPoints": {"type": "MultiPoint", "crs": {"type": "name", "properties":
    {"name": "epsg:4326"}}, "coordinates": [[ -73.976266, 40.788717], [
-73.973562, 40.792193], [ -73.971802, 40.794630]]}
}
```

## 例: 終点の HTTP POST ペイロード

```
{
  "endPoints": {"type": "MultiPoint", "crs": {"type": "name", "properties":
    {"name": "epsg:4326"}}, "coordinates": [[ -73.976266, 40.788717], [
-73.973562, 40.792193], [ -73.971802, 40.794630]]}
}
```

### 一時更新

この優先設定セットを使用して、要求ごとに一時更新を設定することができます。例えば、サーバーがすべての幹線道路タイプを回避するよう要求することができます。要求ごとに 1 つ以上の更新を含めることができます。速度更新の場合、正の速度値は速度の増加、負の速度値は速度の減少です。次は、一時更新タイプの説明です。

## 更新タイプ

## 説明

## point

ポイントの更新は、対応するポイント (緯度、経度) に適用される変更です。特定のポイントに対して、ポイントの除外、ポイントの速度の設定、値または割合によるポイントの速度の変更 (増加または減少) を行うことができます。次のいずれかの更新タイプを使用します。

- percentage** 速度を増加 (正の値) または減少 (負の値) させる割合を指定してポイントの速度の増加を定義する速度更新です。
- speed** 速度単位と新しい速度を指定してポイントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。
- speedAdjustment** 速度 (単位と値) の変更を指定してポイントの速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。
- exclude** 指定されたポイントをルート計算から除外する文字列値です。ポイントを除外するには、ポイントを指定して、**Y** と定義した **exclude** パラメータを含める必要があります。有効な値は **Y** (はい) と **N** (いいえ) です。

更新タイプ

説明

---

segmentID



## 更新タイプ

## 説明

セグメントの更新は、対応するセグメント ID に適用される変更です。特定のセグメントに対して、セグメントの除外、セグメントの速度の設定、値または割合によるセグメントの速度の変更 (増加または減少)、セグメントの道路タイプの変更を行うことができます。次のいずれかの更新タイプを使用します。

<b>percentage</b>	速度を増加 (正の値) または減少 (負の値) させる割合を指定して <code>segmentID</code> の速度の増加を定義する速度更新です。
<b>speed</b>	速度単位と新しい速度を指定して <code>segmentID</code> の新しい速度を定義する速度更新です。速度更新の場合、速度単位には <code>kph</code> (キロメートル/時)、 <code>mph</code> (マイル/時)、 <code>mps</code> (メートル/秒)、 <code>mtpm</code> (メートル/分) のいずれかの値を指定できます。
<b>speedAdjustment</b>	速度 (単位と値) の変更を指定して <code>segmentID</code> の速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。速度更新の場合、速度単位には <code>kph</code> (キロメートル/時)、 <code>mph</code> (マイル/時)、 <code>mps</code> (メートル/秒)、 <code>mtpm</code> (メートル/分) のいずれかの値を指定できます。
<b>exclude</b>	指定された <code>segmentID</code> をルート計算から除外する文字列値です。 <code>segmentID</code> を除外するには、 <code>segmentID</code> を指定して、 <code>Y</code> と定義した <code>exclude</code> パラメータを含める必要があります。有効な値は <code>Y</code> (はい) と <code>N</code> (いいえ) です。
<b>roadType</b>	<p>ルート計算に使用するセグメントの道路タイプの値を変更する文字列値です。</p> <p><code>roadType</code> には、次の種類があります。</p> <ul style="list-style-type: none"> <li>• 進入路</li> <li>• 裏道路</li> <li>• 接続道路</li> <li>• フェリー</li> <li>• 小道</li> <li>• 侵入制限道路 (密集都市部)</li> <li>• 侵入制限道路 (農村部)</li> <li>• 侵入制限道路 (郊外)</li> <li>• 侵入制限道路 (都市部)</li> <li>• 地方道路 (密集都市部)</li> <li>• 地方道路 (農村部)</li> <li>• 地方道路 (郊外)</li> <li>• 地方道路 (都市部)</li> <li>• 主要地方道路 (密集都市部)</li> <li>• 主要地方道路 (農村部)</li> <li>• 主要地方道路 (郊外)</li> <li>• 主要地方道路 (都市部)</li> </ul>

## 更新タイプ

## 説明

- 
- 幹線道路 (密集都市部)
  - 幹線道路 (農村部)
  - 幹線道路 (郊外)
  - 幹線道路 (都市部)
  - 補助地方道路 (密集都市部)
  - 補助地方道路 (農村部)
  - 補助地方道路 (郊外)
  - 補助地方道路 (都市部)
  - 一般道路 (密集都市部)
  - 一般道路 (農村部)
  - 一般道路 (農村部)
  - 一般道路 (都市部)
  - 主要高速道路 (密集都市部)
  - 主要高速道路 (農村部)
  - 主要高速道路 (郊外)
  - 主要高速道路 (都市部)
  - 出入路 (密集都市部)
  - 出入路 (侵入制限道路)
  - 出入路 (幹線道路)
  - 出入路 (主要高速道路)
  - 出入路 (農村部)
  - 出入路 (一般高速道路)
  - 出入路 (都市部)
  - 出入路 (郊外)
  - 一般高速道路 (密集都市部)
  - 一般高速道路 (農村部)
  - 一般高速道路 (郊外)
  - 一般高速道路 (都市部)
-

## 更新タイプ

## 説明

## 道路タイプ

道路タイプの更新は、対応する道路タイプに適用される変更です。特定の道路タイプに対して、道路タイプの速度の設定、値または割合による道路タイプの速度の変更 (増加または減少) を行うことができます。次のいずれかの更新タイプを使用します。

- percentage** 速度を増加 (正の値) または減少 (負の値) させる割合を指定して道路タイプの速度の増加を定義する速度更新です。
- speed** 速度単位と新しい速度を指定して道路タイプの新しい速度を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。
- speedAdjustment** 速度 (単位と値) の変更を指定して道路タイプの速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

## 例: 一時更新の HTTP POST ペイロード

```
{
  "transientUpdates": [
    {
      "segmentID": "7e3396fc:151186f",
      "updates": [
        { "percentage": 26.0 }
      ]
    },
    {
      "point": { "type": "Point",
        "crs": {
          "type": "name",
          "properties": {
            "name": "epsg:4326"
          }
        }
      },
      "coordinates": [
        -73.972776,
        40.795076
      ]
    }
  ]
}
```

```

    },
    "updates": [
      { "speedAdjustment" : { "velocity": 5, "velocityUnit": "kph"} }
    ]
  },
  {
    "roadType": "major road dense urban",
    "updates": [
      { "speed": { "velocity": 25, "velocityUnit": "kph"} }
    ]
  }
]
}

```

### 車両規制

車両規制は、ルートに沿って移動する特定車両の動作と属性をガイドするルーティングエンジンへの規則で構成されます。指定された車両の属性(高さ、幅、長さ、重量など)、および道路網に見られる車両規制の属性に応じて、特定の車両に、あるセグメント上を経由させることができるかどうか決定されます。道路網に車両規制の属性がない場合、入力制限パラメータは、結果的なルートに影響を与えません。

以下に示すのは、車両規制の一連のパラメータです。

#### オプション

#### 説明

##### looseningBarrierRestrictions

ルートを決定する際、通行規制を除去するかどうかを指定します。このような規制は、地方の条例によって特定車両がセグメントの通行を禁止されている場合や、必要となるときのみ(ラストマイルアクセスや地方の配達など)特定車両がそのセグメントで許可されている場合がほとんどです。通行規制が除去されたルートは、通行規制のないルートより短い速い場合でも、引き続きルートコストが高くなります。

## オプション

## 説明

## vehicleAttributes

ルートを決定する際、タイプ、高さ、重量、長さ、または幅に基づいて規制される車両の詳細を指定します。特定車両は、ショートトレーラーから長い3連結トレーラーまでさまざまなタイプに分かれています。車両規制属性は、車両タイプごとにまとめられます。つまり、あるセグメントに対して1つの車両タイプを優先すると共に、同じセグメントで別の車両タイプを規制することが完全に可能です。次のタイプの車両情報を使用します。

- vehicleType** ALL または次のいずれかのタイプの車両を選択します:  
STRAIGHT、SEMI\_TRAILOR、STANDARD\_DOUBLE、INTERMEDIATE\_DOUBLE、LONG\_DOUBLE、TRIPLE、OTHER\_LONG\_COMBINATION\_VEHICLE。
- weight** 車両の最大重量を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。重量の単位は、kg、lb、mt、tです。
- height** 車両の最大高を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。高さの単位は、ft、yd、mi、m、kmです。
- length** 車両の最大長を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。長さの単位は、ft、yd、mi、m、kmです。
- width** 車両の最大幅を指定します。ルートを決定する際、この値を超えるすべての車両は規制されます。幅の単位は、ft、yd、mi、m、kmです。

注：重量/高さ、または長さ/幅のいずれかを、対応する単位と共に指定する必要があります。

## 例

## 車両規制なし

リクエスト:

```
HTTP GET
/rest/Spatial/erm/databases/US_CVR.json?q=routeCostMatrix&startPoints=-74.7221203,42.9737073,eps:4326&endPoints=-74.6671887,42.8097083,eps:4326
```

レスポンス:

```
{
  "matrix": [{
    "distance": 40025,
```

```

    "distanceUnit": "m",
    "endPoint": {
      "type": "Point",
      "coordinates": [-74.6671887, 42.8097083]
    },
    "startPoint": {
      "type": "Point",
      "coordinates": [-74.7221203, 42.9737073]
    },
    "time": 36.57,
    "timeUnit": "min"
  }
}

```

## 車両規制あり

リクエスト:

```

HTTP GET
/rest/Spatial/erm/databases/US_CVR.json?q=routeCostMatrix&startPoints=-74.7221203,42.9737073,eps:4326&
endPoints=-74.6671887,42.8097083,eps:4326

```

車両規制の HTTP POST ペイロード

```

{
  "cvr":{
    "looseningBarrierRestrictions":"n","vehicleAttributes":{
      "vehicleType":"ALL","heightUnit":"meter","height":"4","weightUnit":"Kilogram","weight":"40000"
    }
  }
}

```

レスポンス:

```

{
  "matrix": [{
    "distance": 44933,
    "distanceUnit": "m",
    "endPoint": {
      "type": "Point",
      "coordinates": [-74.6671887, 42.8097083]
    },
    "startPoint": {
      "type": "Point",
      "coordinates": [-74.7221203, 42.9737073]
    },
    "time": 37.48,
  }
}

```

```

    "timeUnit": "min"
  }]
}

```

CVR ありでは距離と時間の値が変化しています。

### 道路タイプの優先度

ルート決定時にさまざまな道路に与える優先順位を指定します。次は、道路タイプの優先度オプションの説明です。

オプション	説明
高	この道路タイプを他の道路タイプよりも優先します。
中	この道路タイプに他の道路タイプと同じ優先設定を与えます。道路タイプに優先設定を指定しないと、デフォルトとして "中" が使われます。
低	他の道路タイプをこの道路タイプよりも優先します。
除外	この道路タイプを可能な限りルートで使用しません。特定の道路タイプを道順から常に除外できるとは限りません。状況によっては、回避対象の道路タイプの代わりになるものが非常に貧弱なために回避対象の道路タイプを使うルートが選ばれることがあります。また、始点または終点がセグメントの横にあって、そのセグメントの道路タイプが回避されている場合も、そのセグメントが使われます。

### 例: 道路タイプの優先度の HTTP POST ペイロード

```

{
  "roadTypesPriority": {
    "RoadType.MajorRoadDenseUrban": "High",
    "RoadType.LimitedAccessDenseUrban": "Low",
    "RoadType.LimitedAccessRural": "Medium",
    "RoadType.PrimaryHighwayUrban": "Avoid"
  }
}

```

## GetSegmentData

### 説明

GetSegmentData サービスは、ポイントまたはセグメント ID のセグメント情報を返します。ポイントを指定すると、最も近いルート セグメントが返されます。セグメント ID を指定すると、指定されたルート セグメントのルート データが返されます。

注： REST サービスからの応答は JSON 形式です。要求の GET URL に無効なクエリ パラメータが含まれていたり、POST のペイロードが無効である場合は、累積的なエラー応答が JSON 配列に返されます。応答 JSON の value ノードは非推奨です。エラーの確認には、errors ノードを使用してください。

### HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。HTTP GET 要求は、ポイントにおけるセグメントデータを返す場合と、セグメント ID のセグメント データを返す場合で異なります。

指定されたポイントにおけるセグメントのデータを返す:

```
HTTP GET
/rest/Spatial/erm/databases/dbsource/segments.json?point=x,y,srsName&query_parameters
```

指定されたセグメントのデータを返す:

```
HTTP GET
/rest/Spatial/erm/databases/dbsource/segments/segmentID.json?query_parameters
```

ここで、**dbsource** は、ルート用のデータを含むデータベースの名前です。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。**segmentID** は、データを返すセグメント識別子です。

### クエリ パラメータ

この操作には、以下のクエリ パラメータが使用できます。



パラメータ	タイプ	必須	説明
destinationSrs	文字列	いいえ	セグメントデータと結果のジオメトリを返す座標系。既定値は使用するデータの座標系です。
distanceUnit	文字列	いいえ	距離を返すときの単位。既定値はm(メートル)です。使用可能な値:m(メートル)、km(キロメートル)、yd(ヤード)、ft(フィート)、mi(マイル)。
時間単位	文字列	いいえ	時間を返すときの単位。既定値はmin(分)です。使用可能な値:min(分)、msec(ミリ秒)、s(秒)、h(時間)。
velocityUnit	文字列	いいえ	速度を返すときの単位。既定値はmph(マイル/時)です。使用可能な値:mph(マイル/時)およびkph(キロメートル/時)。
angularUnit	文字列	いいえ	回転角度を返すときの単位。既定値はdeg(度)です。使用可能な値:deg(度)、rad(ラジアン)、minute(分)、sec(秒)、grad(グラード)。
segmentGeometryStyle	文字列	いいえ	<p>ルートのセグメントを表すジオメトリの形式。デフォルトは <b>None</b> です。セグメント ジオメトリを返す必要がある場合は、このパラメータを指定します。道順を指定するときのオプションは次のとおりです。</p> <p><b>None</b> セグメントのジオメトリ表現を返します。指定されない場合の既定値です。</p> <p><b>End</b> ルートの各セグメントと終点を <b>LineString</b> で返します。</p> <p><b>All</b> 各セグメントとすべての形状ポイントを <b>LineString</b> で返します。<b>LineString</b> はマップ上でオーバーレイとして使用できます。</p>
バージョン	文字列	いいえ	GetSegmentData REST サービスのバージョンを指定します。有効な値は 1 と 2 です。version のデフォルト値は 1 です。

## 例

ポイントを指定するセグメント データを返します。

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase/segments.json?
point=-77,38,epsg:4326&segmentGeometryStyle=all
```

## 応答

```
[{
  "segmentID": "aa18eb33:1b7bbe",
  "primaryName": "VA-631",
  "primaryNameLanguage": "en",
  "alternateNames": [{
    "alternateName": "Lloyds Rd",
    "language": "en"
  }],
  {
    "alternateName": "VA-631",
    "language": "en"
  }],
  "segmentLength": 4.954,
  "segmentLengthUnit": "mi",
  "timeTaken": 5.9333,
  "timeUnit": "min",
  "turnAngle": 0.0,
  "turnAngleUnit": "deg",
  "compassDirection": "",
  "speedOfTravel": 49.9955,
  "speedOfTravelUnit": "mph",
  "roadType": "major road rural",
  "segmentDirection": "bidirectional",
  "startJunctionType": "",
  "endJunctionType": "Other",
  "isRoundabout": false,
  "isTollRoad": false,
  "geometry": {
    "type": "LineString",
    "crs": {
      "type": "name",
      "properties": {
        "name": "epsg:4326"
      }
    }
  },
  "coordinates": [[...]]
},
{
  "segmentID": "46ed0e49:d9a7dc",
  "primaryName": "VA-631",
  "primaryNameLanguage": "en",
  "alternateNameList": [{
```

```

    "alternateName": "Lloyds Rd",
    "language": "en"
  }],
  "segmentLength": 1.198,
  "segmentLengthUnit": "mi",
  "timeTaken": 1.433,
  "timeUnit": "min",
  "turnAngle": 0.0,
  "turnAngleUnit": "degree",
  "compassDirection": "",
  "speedOfTravel": 49.9955,
  "speedOfTravelUnit": "mph",
  "roadType": "major road rural",
  "segmentDirection": "bidirectional",
  "startJunctionType": "Other",
  "endJunctionType": "",
  "isRoundabout": false,
  "isTollRoad": false,
  "pointsInSegment": {
    "type": "LineString",
    "crs": {
      "type": "name",
      "properties": {
        "name": "epsg:4326"
      }
    }
  },
  "coordinates": [[...]]
}
]]

```

セグメント ID を指定するセグメント データを返します。

```

http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase/segments/aa18eb33:1b7bbe.json?
distanceUnits=mi

```

応答

```

[ {
  "segmentID": "aa18eb33:1b7bbe",
  "primaryName": "VA-631",
  "primaryNameLanguage": "en",
  "alternateNames": [ {
    "alternateName": "Lloyds Rd",
    "language": "en"
  } ],
  {
    "alternateName": "VA-631",
    "language": "en"
  } ],
  "segmentLength": 4.954,

```

```

"segmentLengthUnit": "mi",
"timeTaken": 5.9333,
"timeUnit": "min",
"turnAngle": 0.0,
"turnAngleUnit": "deg",
"compassDirection": "",
"speedOfTravel": 49.9955,
"speedOfTravelUnit": "mph",
"roadType": "major road rural",
"segmentDirection": "bidirectional",
"startJunctionType": "",
"endJunctionType": "Other",
"isRoundabout": false,
"isTollRoad": false
}

```

### バージョン固有のエラー応答

リクエストに無効なパラメータ値 (ポイントの欠落など) を入力すると、入力したバージョンによって異なるエラー応答が返ります。バージョンが1の場合、値とエラーが返りますが、バージョンが2の場合、応答にはエラーのみが含まれます。

- version が 1 の場合のリクエスト:

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?version=1
```

- レスポンス:

```

{
  "value": "Point cannot be empty.",
  "errors": [
    {
      "errorCode": 4139,
      "userMessage": "Point cannot be empty."
    }
  ]
}

```

- version が 2 の場合のリクエスト:

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?version=2
```

- レスポンス:

```

{"errors": [
  {
    "errorCode": 4139,
    "userMessage": "Point cannot be empty."
  }
]}

```

```

}
]
}

```

## GetTravelBoundary

### 説明

**GetTravelBoundary** は、ロケーションから一定の時間または距離で、車両または徒歩で到達可能なエリアの境界を示します。この機能では、時間到達圏または距離到達圏の計算に基づくポリゴンを取得します。時間到達圏とは、始点から一定時間内に道路を通過して到達できるエリアを表すポリゴンまたはポイントの集まりです。距離到達圏とは、始点から一定距離の移動で到達できるエリアを表すポリゴンまたはポイントの集まりです。**GetTravelBoundary** 操作 (到達圏検索とも呼ばれます) は、始点、単位 (距離または時間)、および 1 つ以上のコストを入力として受け取り、生成された到達圏 (ポリゴン) を返します。コストとは、到達圏計算に使用する時間または距離の値を意味します。また、複数のコストを入力として提供することもできます。複数のコストの場合、コストはカンマ区切り文字列として提供することもできます。

注: REST サービスからの応答は JSON 形式です。要求の GET URL に無効なクエリパラメータが含まれていたり、POST のペイロードが無効である場合は、累積的なエラー応答が JSON 配列に返されます。応答 JSON の value ノードは非推奨です。エラーの確認には、errors ノードを使用してください。

### HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。HTTP GET は、その他の JSON ペイロードを必要としないすべての到達圏境界に使用されます (周辺移動速度の変更)。

```

HTTP GET
/rest/Spatial/erm/databases/dbsource.json?q=travelBoundary&query_parameters

```

ここで、**dbsource** は、ルート用のデータを含むデータベースの名前です。Enterprise Routing モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### HTTP POST URL 形式

HTTP POST リクエストには次の形式を使用します。

```

HTTP POST:

```

```
/rest/Spatial/erm/databases/dbsource.json?q=travelBoundary&query_parameters
POST BODY: Content-Type:application/json {Route Data}
```

Route Data は、道路タイプの周辺移動速度を含む計算で使用される追加のルート情報を表す POST Json 本体 (Content-Type: application/json) です。これらのオプションの詳細については、[GetTravelBoundary HTTP POST オプション \(156ページ\)](#) を参照してください。

### クエリ パラメータ

この演算は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
point	文字列	はい	到達圏境界を計算するときの始点を x, y, coordSys 形式で指定します。例: -74.2,40.8,EPSG:4326
コスト	Double	はい	距離または時間のコストを、指定されたコスト単位で表して指定します (小数値が使用できます)。例えば、指定された単位がマイルの場合に、このフィールドに 10 を指定すると、10 マイル移動するための到達圏境界が計算されます。また、複数のコストを指定することも可能で、値をカンマ区切り文字列として指定します。指定されたすべてのコストに対して個別に到達圏境界が返されます。複数のコストを指定した場合、応答にはコストとその応答に関連付けられているコスト単位が含まれます。
costUnit	文字列	はい	到達圏境界の計算に使用される計測単位のタイプを指定します。使用可能な距離値は以下のとおりです。 <ul style="list-style-type: none"> <li>• m (メートル)</li> <li>• km (キロメートル)</li> <li>• yd (ヤード)</li> <li>• ft (フィート)</li> <li>• mi (マイル)</li> </ul> 使用可能な時間値は以下のとおりです。 <ul style="list-style-type: none"> <li>• min (分)</li> <li>• msec (ミリ秒)</li> <li>• s (秒)</li> <li>• h (時)</li> </ul>

パラメータ	タイプ	必須	説明
maxOffroadDistance	Double	いいえ	<i>maxOffroadDistanceUnit</i> を使用して道路以外を移動できる距離の最大値。道路以外とは、私道や進入路などのことです。例えば、道路外距離の最大値を 1 マイルと指定すると、道路から 1 マイル以上離れた地点は、到達圏境界に含まれません。道路外距離の最大値として 0 を指定すると、道路から少しでも離れた地点は到達圏境界に含まれません。道路以外を移動する際の速度は、周辺移動速度オプションによって指定します。
maxOffroadDistanceUnit	文字列	いいえ	<i>maxOffroadDistance</i> を定義する距離単位を指定します。また、このパラメータを定義する場合も、 <i>maxOffroadDistance</i> を定義する必要があります。使用可能な距離値は以下のとおりです。 <ul style="list-style-type: none"> <li>• m (メートル)</li> <li>• km (キロメートル)</li> <li>• yd (ヤード)</li> <li>• ft (フィート)</li> <li>• mi (マイル)</li> </ul>
destinationSrs	文字列	いいえ	到達圏境界ジオメトリを返す座標系を指定します。デフォルトは、使用するデータの座標系 (例えば、 <i>epsg:4326</i> ) です。
majorRoads	Boolean	いいえ	計算時に、すべての道路を含めるか、幹線道路のみを含めるかを指定します。幹線道路のみを含めるように指定した場合は、パフォーマンスは向上しますが精度は低くなる場合があります。デフォルトは <b>true</b> です。
returnHoles	Boolean	いいえ	境界内の到達不可能領域 (穴) を返すかどうかを指定します。これは、道順に従って適切な時間または距離で到達できない、主要境界内に存在する領域 (穴) を指します。デフォルトは <b>false</b> です。
returnIslands	Boolean	いいえ	境界外の到達可能領域 (島) を返すかどうかを指定します。これは、適切な時間または距離で到達できる、主要境界の外にある小さな領域です。デフォルトは <b>false</b> です。

パラメータ	タイプ	必須	説明
simplificationFactor	Integer	いいえ	返す必要があるオリジナルのポイントの割合 (結果のジオメトリの複雑さを維持する割合) を指定します。0.0 ~ 1.0 (0.0 は除くが 1.0 は含む) の間の数値が指定できます。値が大きいほど複雑さは増します。つまり、1.0 は最も複雑であることを意味します。既定値は 0.5 です。
bandingStyle	文字列	いいえ	<p>結果に使用する境界線表示形式を指定します。境界形式は、複数のコストに基づいて表示できる複数の到達圏の境界のタイプです。境界形式は、次の形式で返すことができます。</p> <p><b>Donut</b> 境界は、内側のより小さな境界をくり抜く方法で確定されます。これはデフォルトの方法です。</p> <p><b>Encompassing</b> 境界は、他の境界の影響を受けず単独で確定されます。</p>
historicTrafficTimeBucket	文字列	いいえ	<p>ルーティング計算において、旅行速度履歴を使用するかどうかを指定します。これらの速度は、1日のうちの異なる時間帯に基づきます。この機能を使用するには、旅行速度履歴がデータに含まれている必要があります。各国/地域のデータに対する時間帯定義は同じです。時間帯に対する速度は異なる場合があります。次のオプションがあります。</p> <p><b>None</b> デフォルト値です。計算において、旅行速度履歴は使用しません。代わりに速度の平均値を使用します。</p> <p><b>AMPeak</b> 午前のピーク速度でルート进行計算します。午前ピークの時間帯は、1日のうちの 07:00 ~ 10:00 です。</p> <p><b>PMPeak</b> 午後のピーク速度でルート进行計算します。午後ピークの時間帯は、1日のうちの 16:00 ~ 19:00 です。</p> <p><b>OffPeak</b> オフピーク (日中) 速度でルート进行計算します。オフピークの時間帯は、1日のうちの 10:00 ~ 16:00 です。</p> <p><b>Night</b> 夜間速度でルート进行計算します。夜間の時間帯は、1日のうちの 22:00 ~ 04:00 です。</p>



パラメータ	タイプ	必須	説明
defaultAmbientSpeed	文字列	いいえ	<p>(すべての道路タイプの) 到達圏境界を検索するために使用する、道路以外を移動する際の速度を指定します。到達圏境界計算において道路外の移動の処理を制御するには、道路外を移動する際の速度(周辺移動速度)を指定する必要があります。周辺移動速度は、到達圏境界を表すポリゴンの大きさや形状に影響を与える可能性があります。一般的には、周辺移動速度が速いほど、ポリゴンは大きくなります。例えば、あと 5 分の移動が可能で、周辺移動速度が 15 マイル/時の場合は、到達可能地点まで 1.25 マイルの距離があることとなります。周辺移動速度を 10 マイル/時に落とすと、到達可能地点までの距離は 0.83 マイルとなります。</p> <p>注： defaultAmbientSpeed のデフォルト値: <b>15</b></p>
ambientSpeedUnit	文字列	いいえ	<p>周辺移動速度を計算するための計測単位を指定します。使用可能な速度単位は以下のとおりです。</p> <ul style="list-style-type: none"><li>• MPH (マイル/時)</li><li>• KPH (キロメートル/時)</li><li>• MTPS (メートル/秒)</li><li>• MTPM (メートル/分)</li></ul> <p>注： ambientSpeedUnit のデフォルト値: <b>MPH</b></p>

パラメータ	タイプ	必須	説明
伝搬係数	文字列	いいえ	<p>始点から距離到達圏までの距離を(全種類の道路を対象に)計算するために使用するコストの割合を指定します。伝搬係数を距離到達圏に適用する目的は、周辺移動速度を時間到達圏に適用する場合と同じです。つまり、到達圏境界の計算に与える道路外の移動の影響を、この係数で制御します。伝搬係数は、到達圏境界を表すポリゴンの大きさや形状に影響を与える可能性があります。一般に、伝搬係数が大きいほど、ポリゴンも大きくなります。</p> <p>この係数は距離到達圏に適用します。このプロパティを指定しない場合は、計算にはサーバー設定が使用されます。有効な値は 0.0 から 1.0 までです。</p> <p>注:</p> <ul style="list-style-type: none"> <li>• デフォルトの <code>propagationFactor</code> は <b>0.16</b> です。</li> <li>• このパラメータは POST 本体でも指定できます。GET と POST の両方に同じパラメータが設定されている場合は、POST の値が考慮されます。</li> </ul>
バージョン	文字列	いいえ	<p>GetTravelBoundary REST サービスのバージョンを指定します。有効な値は 1 と 2 です。<code>version</code> のデフォルト値は 1 です。</p>

## 例

単一のコストを持つ到達圏境界。

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?q=travelBoundary&point=-77.092609,38.871256,epsG:4326&costs=5&costUnit=m
```

## 応答

```
{
  "travelBoundary": {
    "costs": [
      {
        "cost": 5,
        "costUnit": "m",
        "geometry": {"type": "MultiPolygon", "coordinates": [[[...]]]}
      }
    ]
  }
}
```

```

]
}
}

```

複数のコストを持つ到達圏境界。

```

http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
q=travelBoundary&point=-77.092609,38.871256,epsg:4326&costs=2,5&costUnit=m

```

応答

```

{
  "travelBoundary": {
    "costs": [
      {
        "cost": 2,
        "costUnit": "m",
        "geometry": "{\"type\":\"MultiPolygon\",\"coordinates\":[[[...]]}"
      },
      {
        "cost": 5,
        "costUnit": "m",
        "geometry": "{\"type\":\"MultiPolygon\",\"coordinates\":[[[...]]}"
      }
    ]
  }
}

```

### バージョン固有のエラー応答

リクエストに無効なパラメータ値 (ポイントが境界の外側にある、など) を入力すると、入力したバージョンによって異なるエラー応答が返ります。バージョンが 1 の場合、値とエラーが返りますが、バージョンが 2 の場合、応答にはエラーのみが含まれます。

- version が 1 の場合のリクエスト:

```

http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?q=travelBoundary
&costs=5&costUnit=min&point=-14.321600,60.662859,epsg:4326&version=1

```

- レスポンス:

```

{
  "value": "Point outside boundaries: (-14.3216,60.662859,0)",
  "errors": [
    {

```

```

    "errorCode": 5008,
    "userMessage": "Point outside boundaries: (-14.3216,60.662859,0)"
  }
]
}

```

- version が 2 の場合のリクエスト:

```

http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?q=travelBoundary
&costs=5&costUnit=min&point=-14.321600,60.662859,epsg:4326&version=2

```

- レスポンス:

```

{
  "errors": [
    {
      "errorCode": 5008,
      "userMessage": "Point outside boundaries: (-14.3216,60.662859,0)"
    }
  ]
}

```

### GetTravelBoundary HTTP POST オプション

#### HTTP POST URL 形式

標準の HTTP GET パラメータのほか、道路タイプの周辺移動速度の変更を指定する HTTP POST ペイロード オプションを要求に追加できます。コンテンツタイプは application/json に設定する必要があります。HTTP POST リクエストには次の形式を使用します。

```

HTTP POST:
/rest/Spatial/erm/databases/dbsource.json?q=travelBoundary&query_parameters
POST BODY: Content-Type:application/json {Route Data}

```

Route Data は、道路タイプの周辺移動速度を含む計算で使用される追加のルート情報を表す POST Json 本体 (Content-Type: application/json) です。

#### 周辺移動速度

この優先設定セットを使用して、要求ごとに周辺移動速度の変更を設定できます。周辺移動速度とは、経路境界を検索するときに道路以外を移動するために使用する、標準データにおける速度

の変更です。道路以外とは、私道や進入路などのことです。次は、周辺移動速度パラメータの説明です。

## パラメータ

## 説明

---

### *DefaultAmbientSpeed*

(すべての道路タイプの) 到達圏境界を検索するために使用する、道路以外を移動する際の速度を指定します。到達圏境界計算において道路外の移動の処理を制御するには、道路外を移動する際の速度(周辺移動速度)を指定する必要があります。周辺移動速度は、到達圏境界を表すポリゴンの大きさや形状に影響を与える可能性があります。一般的には、周辺移動速度が速いほど、ポリゴンは大きくなります。例えば、あと 5 分の移動が可能で、周辺移動速度が 15 マイル/時の場合は、到達可能地点まで 1.25 マイルの距離があることとなります。周辺移動速度を 10 マイル/時に落とすと、到達可能地点までの距離は 0.83 マイルとなります。

注: DefaultAmbientSpeed のデフォルト値: **15**

---

### *AmbientSpeedUnit*

周辺移動速度を計算するための計測単位を指定します。使用可能な速度単位は以下のとおりです。

- MPH (マイル/時)
- KPH (キロメートル/時)
- MTPS (メートル/秒)
- MTPM (メートル/分)

注: AmbientSpeedUnit のデフォルト値: **MPH**

---

パラメータ

説明

---

*AmbientSpeed.RoadType*

## パラメータ

## 説明

道路の種類ごとに、道路外を移動する際の周辺移動速度を指定します。道路タイプとその道路タイプの新しい速度を指定する必要があります。速度は、定義済みの AmbientSpeedUnit に定義されます。道路タイプは、サポートされているすべてのタイプで返すことができます。次の道路タイプを使用できます。

- 進入路
- 裏道路
- 接続道路
- フェリー
- 小道
- 侵入制限道路 (密集都市部)
- 侵入制限道路 (農村部)
- 侵入制限道路 (郊外)
- 侵入制限道路 (都市部)
- 地方道路 (密集都市部)
- 地方道路 (農村部)
- 地方道路 (郊外)
- 地方道路 (都市部)
- 主要地方道路 (密集都市部)
- 主要地方道路 (農村部)
- 主要地方道路 (郊外)
- 主要地方道路 (都市部)
- 幹線道路 (密集都市部)
- 幹線道路 (農村部)
- 幹線道路 (郊外)
- 幹線道路 (都市部)
- 補助地方道路 (密集都市部)
- 補助地方道路 (農村部)
- 補助地方道路 (郊外)
- 補助地方道路 (都市部)
- 一般道路 (密集都市部)
- 一般道路 (農村部)
- 一般道路 (農村部)
- 一般道路 (都市部)
- 主要高速道路 (密集都市部)
- 主要高速道路 (農村部)
- 主要高速道路 (郊外)
- 主要高速道路 (都市部)
- 出入路 (密集都市部)
- 出入路 (侵入制限道路)
- 出入路 (幹線道路)

## パラメータ

## 説明

- 
- 出入路 (主要高速道路)
  - 出入路 (農村部)
  - 出入路 (一般高速道路)
  - 出入路 (都市部)
  - 出入路 (郊外)
  - 一般高速道路 (密集都市部)
  - 一般高速道路 (農村部)
  - 一般高速道路 (郊外)
  - 一般高速道路 (都市部)

## 伝搬係数

伝搬係数の詳細については、[クエリパラメータ \(150ページ\)](#) を参照してください。

注：このパラメータは、getTravelBoundary バージョン 2 が指定されている場合にのみサポートされます。

---



パラメータ

説明

---

道路タイプ

## パラメータ

## 説明

道路タイプに基づいて、道路外を移動する際に使用する **伝搬係数** を指定します。道路タイプと、その道路タイプに対する伝搬係数の新しい値の両方を指定する必要があります。詳細については、「**伝搬係数**」を参照してください。道路タイプは、サポートされているすべてのタイプで返すことができます。次の道路タイプを使用できます。

- 進入路
- 裏道路
- 接続道路
- フェリー
- 小道
- 侵入制限道路 (密集都市部)
- 侵入制限道路 (農村部)
- 侵入制限道路 (郊外)
- 侵入制限道路 (都市部)
- 地方道路 (密集都市部)
- 地方道路 (農村部)
- 地方道路 (郊外)
- 地方道路 (都市部)
- 主要地方道路 (密集都市部)
- 主要地方道路 (農村部)
- 主要地方道路 (郊外)
- 主要地方道路 (都市部)
- 幹線道路 (密集都市部)
- 幹線道路 (農村部)
- 幹線道路 (郊外)
- 幹線道路 (都市部)
- 補助地方道路 (密集都市部)
- 補助地方道路 (農村部)
- 補助地方道路 (郊外)
- 補助地方道路 (都市部)
- 一般道路 (密集都市部)
- 一般道路 (農村部)
- 一般道路 (農村部)
- 一般道路 (都市部)
- 主要高速道路 (密集都市部)
- 主要高速道路 (農村部)
- 主要高速道路 (郊外)
- 主要高速道路 (都市部)
- 出入路 (密集都市部)
- 出入路 (侵入制限道路)

## パラメータ

## 説明

- 出入路 (幹線道路)
- 出入路 (主要高速道路)
- 出入路 (農村部)
- 出入路 (一般高速道路)
- 出入路 (都市部)
- 出入路 (郊外)
- 一般高速道路 (密集都市部)
- 一般高速道路 (農村部)
- 一般高速道路 (郊外)
- 一般高速道路 (都市部)

注: このパラメータは、getTravelBoundary バージョン 2 が指定されている場合にのみサポートされます。

## 除外

到達圏境界計算中に回避する道路タイプのリストをカンマで区切って指定します。これは文字列パラメータです。パラメータの値として道路タイプを指定すると、計算で境界からそのタイプの道路が除外されます。例えば、パラメータの値として有料道路を指定すると、計算された境界から有料道路が除外されます。

gettravelboundary バージョン 1 の HTTP POST ペイロードでの周辺移動速度パラメータの使用例。

```
{
  "DefaultAmbientSpeed": 45,
  "AmbientSpeedUnit": "MPH"

  "AmbientSpeed.RoadType.PrimaryHighwayUrban": 15,
  "AmbientSpeed.RoadType.SecondaryHighwayUrban": 10
}
```

gettravelboundary バージョン 2 の HTTP POST ペイロードでの周辺移動速度パラメータおよび伝搬係数パラメータの使用例。

```
{
  "ambientSpeeds": {
    "defaultAmbientSpeed": 24,
    "ambientSpeedUnit": "MPH",
    "ambientSpeedOverrides": {
      "Primary Highway Urban": ".51",
      "Secondary Highway Urban": ".1"
    }
  },
  "propagationFactors": {
```

```

    "propagationFactor": "1",
    "propagationFactorOverrides": {
      "Primary Highway Urban": ".51",
      "Secondary Highway Urban": ".1"
    }
  }
}
{
  "ambientSpeeds": {
    "ambientSpeedOverrides": {
      "Primary Highway Urban": 25,
      "Secondary Highway Urban": 10
    }
  },
  "propagationFactors": {
    "propagationFactor": "0.2",
    "propagationFactorOverrides": {
      "Primary Highway Urban": "0.51",
      "Secondary Highway Urban": "0.1"
    }
  }
}
}

```

注：REST 要求からの応答は JSON 形式で、返されるジオメトリは GEOJSON 形式です。

### GetTravelBoundary (非推奨)

**重要：**このステージは 12.2 リリースで非推奨になっています。新しいデータフローを作成するときは、代わりに [到達可能エリア](#) ステージを使用してください。

GetTravelBoundary は、ロケーションから一定の時間または距離で、車両または徒歩で到達可能なエリアの境界を示します。この機能では、時間到達圏または距離到達圏の計算に基づくポリゴンを取得します。時間到達圏とは、始点から一定時間内に道路を通過して到達できるエリアを表すポリゴンまたはポイントの集まりです。距離到達圏とは、始点から一定距離の移動で到達できるエリアを表すポリゴンまたはポイントの集まりです。Get Travel Boundary 操作 (到達圏定義とも呼ばれます) は、始点、単位 (距離または時間)、1 つ以上のコスト、およびそれらに関連付けられたタグを入力として受け取り、生成された到達圏境界を返します。コストとは、到達圏計算に使用する時間または距離の量を意味します。タグは、コストを一意に指す文字列で、対応する結果とのマッチングに使用されます。複数のコストを“;”区切り文字列として提供することで、コストを入力として指定できます。

GetTravelBoundary は、Spatial モジュールに含まれています。

注：GetTravelBoundary は、Web サービスとしてのみ利用可能です。GetTravelBoundary を、Java、C++、C、.NET、または COM API を介して利用することはできません。

## リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

## JSON 応答の例

次の例では、JSON 応答を要求します。

```
http://myserver:8080/rest/GetTravelBoundary/results.json?
Data.Latitude=51.5072&Data.Longitude=0.1275&Data.&Data.TravelBoundaryCost=10&
Data.TravelBondaryCostUnits=Kilometers&Option.DataSetResourceName=Routing_db_gb
```

この要求から次のような JSON が返されます。

注：この例では、短縮のため一部のポイントが削除されています。

注：この例の応答では、返されたジオメトリの座標が小数点以下 14 桁の値で表示されています。これは、特に大きなポリゴンや多くのレコードが返される場合に、JSON 応答のペイロードを必要以上に増加させます。また、自己交差などの無効なジオメトリにつながる恐れもあります。桁数を減らした座標値を返すことによってペイロードを抑えるには、次の文を `%Spectrum%\server\bin\wrapper\wrapper.conf` に追加してサーバーを再起動します: `wrapper.java.additional.xx=-Dcom.pb.midev.useprecision=true`。

```
{
  "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": "TravelBoundaryCostUnits",
      "value": "Kilometers"
    }
  ]
}]

```

### XML 応答の例

次の例では、XML 応答を要求します。

```

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

```

この要求から次のような XML が返されます。

注：この例では、短縮のため一部のポイントが削除されています。

```
<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:LineString>
          </ns2:Exterior>
        </ns2:Polygon>
      </ns3:IsoPolygonResponse>
    </ns3:IsoRouteResponse>
  </ns3:output_port>
</ns3:xml.GetTravelBoundaryResponse>
```

```

        <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>

```

## 要求

### 入力データのパラメータ

`GetTravelBoundary` は、コスト、コスト単位、ポイント緯度、およびポイント経度を入力として受け取ります。以下の表に、入力フォーマットとレイアウトに関する情報を示します。

表 11 : `GetTravelBoundary` 入力データ

パラメータ	書式	説明
<code>Data.Latitude</code>	文字列	ポイントの緯度。 <code>Option.CoordinateFormat</code> パラメータで選択されたフォーマットを使って緯度を指定します。
<code>Data.Longitude</code>	文字列	ポイントの経度。 <code>Option.CoordinateFormat</code> パラメータで選択されたフォーマットを使って経度を指定します。
<code>Data.TravelBoundaryCost</code>	文字列	<p>(オプション) <code>Data.TravelBoundaryCostUnits</code> パラメータまたは <code>Option.DefaultTravelBoundaryCostUnits</code> パラメータで指定された単位で表された距離または時間のコスト。例えば、指定された単位がマイルのときにこのフィールドに 10 を指定すると、コストは 10 マイルになります。</p> <p>このフィールドを使って、デフォルト経路境界コストをレコード単位で上書きできます。</p> <p>また、複数のコストを指定することも可能で、値を ";" で区切られた文字列として指定します。指定されたすべてのコストに対して個別に <code>Iso Route Response</code> が返されます。複数のコストを指定した場合、応答にはコストとその応答に関連付けられている <code>costUnits</code> が含まれます。</p>



パラメータ	書式	説明
Data.TravelBoundaryCostUnits	文字列	<p>(オプション)経路境界の計算に使用される計測単位のタイプ。次のいずれかです。</p> <ul style="list-style-type: none"> <li>• Feet</li> <li>• Hours</li> <li>• Kilometers</li> <li>• Meters</li> <li>• Miles</li> <li>• Minutes</li> <li>• Seconds</li> <li>• Yards</li> </ul> <p>このフィールドを使って、デフォルト経路境界コスト単位をレコード単位で上書きできます。</p>

### オプションのパラメータ

入力

表 12 : GetTravelBoundary の入力オプション

パラメータ	説明
Option.DataSetResourceName	<p>検索プロセスで使用するデータが格納されたデータベースの名前。Management Console の [リソース] セクションで定義した有効なルーティング データベース リソース名を使用します。詳細については、『<i>Spectrum™ Technology Platform Spatial ガイド</i>』を参照してください。</p>
Option.CoordinateSystem	<p>入力される座標の座標系。</p> <p>EPSG コードの詳細については、<a href="http://www.spatialreference.org">www.spatialreference.org</a> を参照してください。EPSG に対してサポートされるコードスペースのリストを取得するには、SOAP リクエスト List Supported CoordSys by Code Space を Geometry サービス デモ ページ <code>http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html</code> から送信します。</p>

## パラメータ

## 説明

Option.CoordinateFormat

入力座標の緯度/経度のフォーマットを指定します。

注：このオプションは、緯度/経度座標系を指定した場合のみに使用します。座標系が緯度/経度座標系ではない場合は、座標フォーマットを [小数] に設定します。

次のいずれかです。

<b>Decimal</b>	(90.000000, 180.000000)
<b>DecimalAssumed</b>	(90000000, 180000000)。こちらがデフォルトです。
<b>DegreesMinutesSeconds</b>	(90 00 00N, 180 00 00W)
<b>PreZero</b>	(090000000N, 180000000W)
<b>PreZeroDecimal</b>	(090.000000N, 180.000000W)

Option.DefaultTravelBoundaryCost

コスト単位の値。任意の `double` 型の値 (小数を含む) を指定できます。デフォルトは 10 です。

Option.DefaultTravelBoundaryCostUnits

経路境界の計算に使う測定単位。次のいずれかです。

- Feet
- Hours
- Kilometers
- Meters
- Miles
- Minutes
- Seconds
- Milliseconds
- Yards

## パラメータ

## 説明

Option.historicTrafficTimeBucket

ルーティング計算において、トラフィック速度履歴を使用するかどうかを指定します。これらの速度は、1日のうちの異なるタイム バケットに基づきます。この機能を使用するには、トラフィック速度履歴がデータに含まれている必要があります。各国/地域のデータに対するバケット定義は同じです。バケット値に対する速度は異なる場合があります。次のオプションがあります。

**None** デフォルト値です。計算において、トラフィックデータ履歴は使用しません。代わりに速度の平均値を使用します。

**AMPeak** 午前のピーク速度でルートを計算します。午前ピークのタイム バケットは、1日のうちの 07:00 ~ 10:00 です。

**PMPeak** 午後のピーク速度でルートを計算します。午後ピークのタイム バケットは、1日のうちの 16:00 ~ 19:00 です。

**OffPeak** オフピーク (日中) 速度でルートを計算します。オフピークのタイム バケットは、1日のうちの 10:00 ~ 16:00 です。

**Night** 夜間速度でルートを計算します。夜間のタイム バケットは、1日のうちの 22:00 ~ 04:00 です。

## 出力

表 13 : GetTravelBoundary の出力オプション

## パラメータ

## 説明

Option.ResultType

返す結果のタイプを指定します。次のいずれかです。

**AccessibleNodes** isoChrone 全体を返します。

**Geometry** isoChrone 計算に到達しうる、道路沿いのすべてのポイントを返します。

**StartNodes** 入力される住所で指定される場所を返します。

パラメータ	説明
Option.SimplificationFactor	元のポイントをどの割合で返す必要があるか(結果のポリゴンの基になるポイントの割合)を指定します。
Option.BandingStyle	結果に使用する境界線表示形式を指定します。境界線表示形式は、複数のコストに基づいて表示できる複数の <b>isoChrono</b> または境界線のタイプです。 <b>Donut</b> 境界は、次に小さな境界をくり抜く方法で確定されます。 <b>Encompassing</b> 境界は、他のすべての境界から独立して確定されます。
Option.ReturnHoles	境界内の到達不可能領域を返すかどうかを指定します。これは、道順に従って適切な時間または距離で到達できない、より大きな境界内に存在する領域を指します。 <b>Y</b> 境界内の到達不可能領域を返します。 <b>N</b> 境界内の到達不可能領域を返しません。こちらがデフォルトです。
Option.ReturnIslands	境界外の到達可能領域を返すかどうかを指定します。これは、適切な時間または距離で到達できない、主要境界の外にある小さな領域です。 <b>Y</b> 境界外の到達可能領域を返します。 <b>N</b> 境界外の到達可能領域を返しません。こちらがデフォルトです。

## 経路

経路オプションでは、道路以外を移動する際の移動速度や、経路境界を計算する際に幹線道路のみを走行するかどうかなどを指定します。多くの経路オプションは、周辺移動速度に関連するものです。

表 14 : GetTravelBoundary の経路オプション

パラメータ	説明
Option.MaximumOffRoadDistance	道路以外を移動する距離の最大値を指定します。道路以外とは、私道や進入路などのことです。例えば、道路外距離の最大値を 1 マイルと指定すると、道路から 1 マイル以上離れた地点は、到達圏境界に含まれません。道路外距離の最大値として 0 を指定すると、道路から少しでも離れた地点は到達圏境界に含まれません。道路以外を移動する際の速度は、周辺移動速度オプションによって指定します。
Option.Units	返されるデータの単位を指定します。次のいずれかです。 <ul style="list-style-type: none"><li>• キロメートル (デフォルト)</li><li>• メートル</li><li>• マイル</li></ul>

パラメータ

説明

---

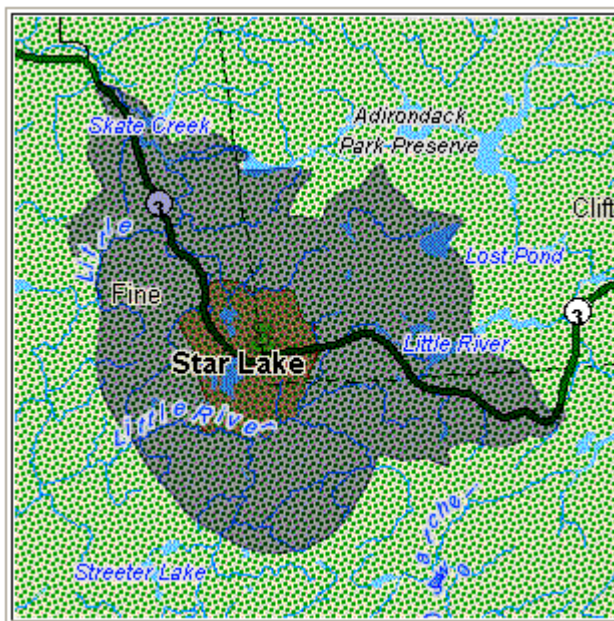
Option.MajorRoads

## パラメータ

## 説明

計算時に、すべての道路を含めるか、幹線道路のみを含めるかを指定します。デフォルトでは、幹線道路が **True** に設定され、**Get Travel Boundary** が計算されます。これによりパフォーマンスが向上しますが、精度は低下する可能性があります。

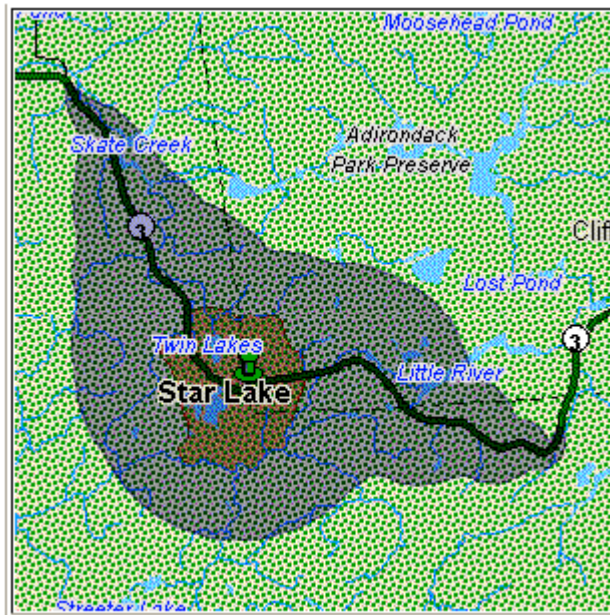
次のマップは、すべての道路を走行する場合の経路境界を表します。



次のマップは、幹線道路のみを走行する場合の経路境界を表します。

## パラメータ

## 説明



次のいずれかです。

- Y** 計算時に、幹線道路のみを含めます。デフォルト
- N** 計算時に、すべての道路を含めます。

## Option.DefaultAmbientSpeed

経路境界を検索するために使用する、道路以外を移動する際の速度を指定します。道路以外とは、私道や進入路などのことです。

このオプションは、Option.DefaultCostUnits パラメータ、または Data.TravelBoundaryCostUnits パラメータで時間値を指定した場合のみ有効です。デフォルト値は 15 です。速度の単位は Option.AmbientSpeedUnit パラメータで指定します。

到達圏境界計算において道路外の移動の処理を制御するには、道路外を移動する際の速度 (周辺移動速度) を指定する必要があります。周辺移動速度は、到達圏境界を表すポリゴンの大きさや形状に影響を与える可能性があります。一般的には、周辺移動速度が速いほど、ポリゴンは大きくなります。例えば、あと 5 分の移動が可能で、周辺移動速度が 15 マイル/時の場合は、到達可能地点まで 1.25 マイルの距離があることとなります。周辺移動速度を 10 マイル/時に落とすと、到達可能地点までの距離は 0.83 マイルになります。

Option.MaximumOffRoadDistance パラメータを使用することにより、道路外を移動する距離を制限できることに注意してください。

注：歩行者の経路境界を計算する場合は、デフォルト周辺移動速度を 3 MPH (5 KPH) に変更することを推奨します。



## パラメータ

## 説明

---

Option.AmbientSpeedUnit

Option.DefaultAmbientSpeed パラメータで指定する値に使用する単位を指定します。

<b>KPH</b>	キロメートル/時
<b>MPH</b>	マイル/時デフォルト
<b>MTPS</b>	メートル/秒
<b>MTPM</b>	メートル/分

---

パラメータ

説明

---

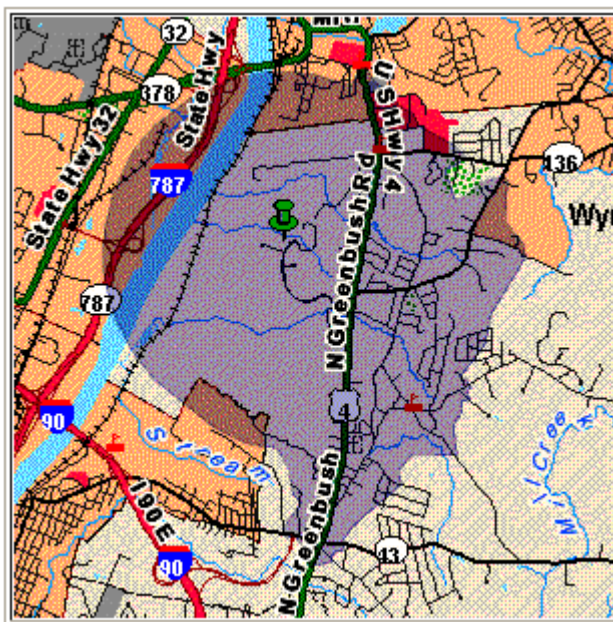
Option.AmbientSpeed.RoadType.<Type>

## パラメータ

## 説明

道路の種類ごとに、道路外を移動する際の周辺移動速度を指定します。道路のタイプに対する周辺移動速度を指定しない場合は、Option.DefaultAmbientSpeed パラメータで指定されたデフォルト周辺移動速度が適用されます。

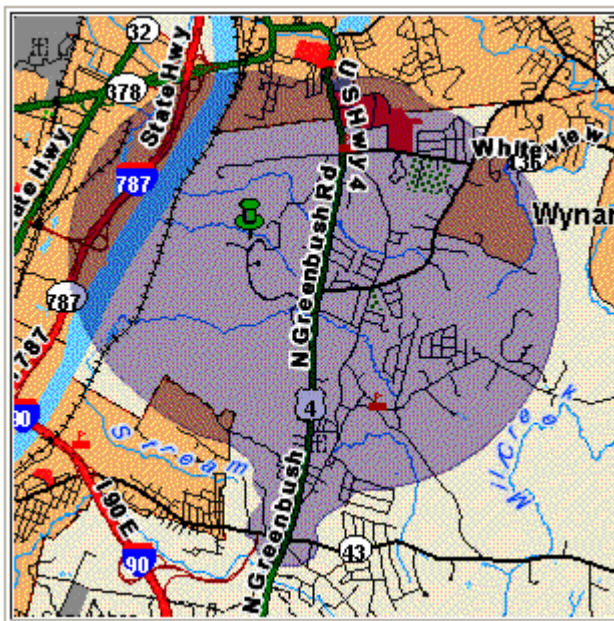
次のマップは、周辺移動速度のオーバーライド値を指定しない場合の経路境界の例を示しています。



比較のために、周辺移動速度のオーバーライド値を指定した場合の、上と同じ経路境界を次のマップに示します。

パラメータ

説明



<Type> には、次の種類があります。

## パラメータ

## 説明

- 
- AccessWay
  - Backroad
  - Connector
  - Ferry
  - Footpath
  - LimitedAccessDenseUrban
  - LimitedAccessRural
  - LimitedAccessSuburban
  - 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
-

## 応答

Get Travel Boundary は、次のフィールドを返します。

表 15 : GetTravelBoundary の出力

応答要素	書式	説明
Status	文字列	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	文字列	失敗の原因 (ある場合)。次のいずれかです。 <ul style="list-style-type: none"> <li>• InsufficientInputData (緯度/経度の欠落)</li> <li>• MalformedInputData (誤った入力フォーマット)</li> <li>• InputOutOfRange (範囲外の入力)</li> <li>• EngineError (エンジン由来のエラー)</li> </ul>
Status.Description	文字列	Status.Code で示される失敗に関する説明。

## PersistentUpdate

### 説明

PersistentUpdate サービスを使用すると、ユーザはネットワークデータの要素を上書きできます。上書きは、道路タイプごと、特定のポイント、または特定のセグメントで実行できます。永続更新は特定のデータ ソースに対してのみ有効で、データ更新後には有効でなくなる場合があります。

永続更新でこのような変更を行うことで、次の操作を実行できます。

- ポイントの除外
- セグメントの除外
- ポイント、セグメント、道路タイプの速度の設定
- ポイント、セグメント、道路タイプの速度の値による変更 (増加または減少)
- ポイント、セグメント、道路タイプの速度の割合による変更 (増加または減少)
- 永続更新の一覧表示

注：永続更新はルーティング データに対してシステム全体で行われる変更であり、すべての更新が永続するため、注意して使用する必要があります。REST サービスからの応答は、正常終了を示すメッセージになります。要求の GET URL に無効なクエリ パラメータが含まれていたり、POST のペイロードが無効である場合は、累積的なエラー応答が JSON 配列に返されます。応答 JSON の value ノードは非推奨です。エラーの確認には、errors ノードを使用してください。

### バージョン固有のエラー応答

リクエストに無効なパラメータ値 (複数の更新など) を入力すると、入力したバージョンによって異なるエラー応答が返ります。バージョンが 1 の場合、値とエラーが返りますが、バージョンが 2 の場合、応答にはエラーのみが含まれます。

- version が 1 の場合のリクエスト:

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
velocity=15.912&velocityUnit=KPH&velocityAdjustment=34&velocityPercentage=56&version=1
```

- レスポンス:

```
{
  "value": "One of either Velocity or SpeedIncrease or SpeedDecrease
is expected.",
  "errors": [
    {
      "errorCode": 3733,
      "userMessage": "One of either Velocity or SpeedIncrease or
SpeedDecrease is expected."
    }
  ]
}
```

- version が 2 の場合のリクエスト:

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase.json?
velocity=15.912&velocityUnit=KPH&velocityAdjustment=34&velocityPercentage=56&version=2
```

- レスポンス:

```
{
  "errors": [
    {
      "errorCode": 3733,
      "userMessage": "One of either Velocity or SpeedIncrease or
SpeedDecrease is expected."
    }
  ]
}
```

## 永続更新のタイプ

永続更新要求タイプとその例については、次のセクションを参照してください。

### ポイントの更新

#### HTTP POST URL 形式

HTTP POST 要求には次の形式を使用します。HTTP POST は、永続更新をポイントに設定するために使用します。

```
HTTP POST:
/rest/Spatial/em/databases/dbsource/persistentUpdates.json?point=x,y,srsName&query_parameters
```

ここで、**dbsource** は、ルート データを更新するデータベースの名前です。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

#### HTTP DELETE URL 形式

HTTP DELETE 要求には次の形式を使用します。HTTP DELETE は、ポイントに対する特定の永続更新を削除するために使用します。

```
HTTP DELETE:
/rest/Spatial/em/databases/dbsource/persistentUpdates.json?point=x,y,srsName&resetType=query_parameters
```

ここで、**dbsource** は、削除する永続更新を含むデータベースの名前です。Enterprise Routing モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### クエリ パラメータ

HTTP POST 操作は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
除外	文字列	いいえ	指定されたポイントをすべてのルート計算から除外します。除外するかどうかの指定は、URL 内にパラメータが存在するかどうかで決まり、パラメータ値では決まりません。



パラメータ	タイプ	必須	説明
速度	文字列	いいえ	新しい速度を指定してポイントの新しい速度を定義する速度更新です。 <b>velocityUnit</b> パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
<b>velocityUnit</b>	文字列	いいえ	<b>velocity</b> または <b>velocityAdjustment</b> の速度の単位 (マイル/時) です。速度更新の場合、速度単位には mph (マイル/時) および kph (キロメートル/時) のいずれかの値を指定できます。デフォルト値は mph です。
<b>velocityAdjustment</b>	文字列	いいえ	速度 (単位と値) の変更を指定してポイントの速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。 <b>velocityUnit</b> パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
<b>velocityPercentage</b>	Integer	いいえ	速度を増加 (正の値) または減少 (負の値) させる割合を指定してポイントの速度の増加を定義する速度更新です。

### リセット パラメータ

HTTP DELETE 操作は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
<b>resetType</b>	文字列	いいえ	ポイントの更新のタイプをリセット (取り消し) します。  <b>speed</b> 特定のポイントの速度更新をリセットします。  <b>exclude</b> 特定のポイントの除外をリセットします。

## 例

### ポイントを除外する (HTTP POST)

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase/persistentUpdates.json?point=-73.6,43.5,epsq:4326&exclude=true
```

### ポイント除外永続更新を削除する (HTTP DELETE)

```
http://www.pb.com/rest/Spatial/erm/databases/usroutedatabase/persistentUpdates.json?point=-73.6,43.5,epsq:4326&resetType=exclude
```

## セグメントの更新

### HTTP POST URL 形式

HTTP POST 要求には次の形式を使用します。HTTP POST は、永続更新をセグメントに設定するために使用します。

```
HTTP POST:  
/rest/Spatial/erm/databases/dbsource/persistentUpdates/segments/segment_id.json?query_parameters
```

ここで、**dbsource** は、ルート データを更新するデータベースの名前で、**segment\_id** は、更新するセグメントの識別子です。**Spatial** モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。HTTP GET は、セグメントの永続更新のリストを返すために使用します。

```
HTTP GET:  
/rest/Spatial/erm/databases/dbsource/persistentUpdates/segments/segment_id.json  
  
or  
  
HTTP GET:  
/rest/Spatial/erm/databases/dbsource/persistentUpdates/segments.json?segments=segment_id
```

ここで、**dbsource** は、永続更新を返すデータベースの名前で、**segment\_id** は、更新を返すセグメントです。

注：最初の形式は、1つのセグメントのみに対して永続更新を返すために使用します。2つめの形式は、複数またはすべてのセグメントに対して永続更新を返すために使用します。複数のセグメントを指定する場合は、セグメント ID をカンマで区切ります。すべてのセグメントを指定する場合は、パラメータ "segments=" を空にします。以下の例を参照してください。

### HTTP DELETE URL 形式

HTTP DELETE 要求には次の形式を使用します。HTTP DELETE は、セグメントに対する特定の永続更新を削除するために使用します。

```
HTTP DELETE:
/rest/Spatial/em/databases/dbsource/persistentUpdates/segments?segmentID=segment_id&resetType=query_parameters
```

ここで、**dbsource** は、データベースの名前で、**segment\_id** は、削除する永続更新を含む更新対象セグメントの識別子です。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### クエリ パラメータ

HTTP POST 操作は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
除外	文字列	いいえ	指定されたセグメントをすべてのルート計算から除外します。除外するかどうかの指定は、URL 内にパラメータが存在するかどうかで決まり、パラメータ値では決まりません。
速度	文字列	いいえ	新しい速度を指定してセグメントの新しい速度を定義する速度更新です。 <b>velocityUnit</b> パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
<b>velocityUnit</b>	文字列	いいえ	<b>velocity</b> または <b>velocityAdjustment</b> の速度の単位です。速度更新の場合、速度単位には mph (マイル/時) および kph (キロメートル/時) のいずれかの値を指定できます。デフォルト値は mph です。

パラメータ	タイプ	必須	説明
<i>velocityAdjustment</i>	文字列	いいえ	速度(単位と値)の変更を指定してセグメントの速度の変化を定義する速度更新です。速度値は増加(正の値)または減少(負の値)させることができます。 <i>velocityUnit</i> パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
<i>velocityPercentage</i>	Integer	いいえ	速度を増加(正の値)または減少(負の値)させる割合を指定してセグメントの速度の増加を定義する速度更新です。

パラメータ	タイプ	必須	説明
道路タイプ	文字列	いいえ	

パラメータ	タイプ	必須	説明
			<p>セグメントの新しい道路タイプを定義する更新です。道路タイプには、次の種類があります。</p> <ul style="list-style-type: none"> <li>• access way</li> <li>• back road</li> <li>• connector</li> <li>• ferry</li> <li>• footpath</li> <li>• limited access dense urban</li> <li>• limited access rural</li> <li>• limited access suburban</li> <li>• limited access urban</li> <li>• local road dense urban</li> <li>• local road rural</li> <li>• local road suburban</li> <li>• local road urban</li> <li>• major local road dense urban</li> <li>• major local road rural</li> <li>• major local road suburban</li> <li>• major local road urban</li> <li>• major road dense urban</li> <li>• major road rural</li> <li>• major road suburban</li> <li>• major road urban</li> <li>• minor local road dense Urban</li> <li>• minor local road rural</li> <li>• minor local road suburban</li> <li>• minor local road urban</li> <li>• normal road dense urban</li> <li>• normal road rural</li> <li>• normal road rural</li> <li>• normal road urban</li> <li>• primary highway dense urban</li> <li>• primary highway rural</li> <li>• primary highway suburban</li> <li>• primary highway urban</li> <li>• ramp dense urban</li> <li>• ramp limited access</li> <li>• ramp major road</li> <li>• ramp primary highway</li> <li>• ramp rural</li> <li>• ramp secondary highway</li> <li>• ramp urban</li> <li>• ramp suburban</li> <li>• secondary highway dense urban</li> <li>• secondary highway rural</li> </ul>

パラメータ	タイプ	必須	説明
			<ul style="list-style-type: none"> <li>secondary highway suburban</li> <li>secondary highway urban</li> </ul>

### リセット パラメータ

HTTP DELETE 操作は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
<i>resetType</i>	文字列	はい	<p>セグメントの更新のタイプをリセット (取り消し) します。</p> <p><b>speed</b> 特定のセグメントの速度更新をリセットします。</p> <p><b>exclude</b> 特定のセグメントの除外をリセットします。</p> <p><b>roadType</b> 特定のセグメントの道路タイプをリセットします。</p>

### 例

セグメントを除外する (HTTP POST)

```
http://<server>:<port>/rest/Spatial/em/databases/US_NE/persistentUpdates/segments/9f5c5a5a:5174e2.json?exclude=true
```

1つのセグメントに対する更新のリストを返す (HTTP GET)

```
http://<server>:<port>/rest/Spatial/em/databases/US_NE/persistentUpdates/segments/efed6c1:a59ac5.json?velocityUnit=kph
```

US\_NE ルーティング データベース リソースのすべてのセグメント更新のリストを返す (HTTP GET)

```
http://<server>:<port>/rest/Spatial/em/databases/US_NE/persistentUpdates/segments.json?segments=
```

## 複数のセグメントに対する更新のリストを返す (HTTP GET)

```
http://<server>:<port>/rest/Spatial/erm/databases/US_NE/persistentUpdates/segments.json?segments=27e20762;4718d9,7e3396fc;14c9c2c
```

## セグメント速度永続更新を削除する (HTTP DELETE)

```
http://<server>:<port>/rest/Spatial/erm/databases/US_NE/persistentUpdates/segments?segmentID=9f5c5a5a;5174e2&resetType=speed
```

## 道路タイプの更新

### HTTP POST URL 形式

HTTP POST 要求には次の形式を使用します。HTTP POST は、永続更新を道路タイプに設定するために使用します。

```
HTTP POST:  
/rest/Spatial/erm/databases/dbsource/persistentUpdates/roadTypes/roadtype.json?query_parameters
```

ここで、**dbsource** は、ルート データを更新するデータベースの名前で、**roadtype** は、更新する道路のタイプです。**Spatial** モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。HTTP GET は、道路タイプの永続更新のリストを返すために使用します。

```
HTTP GET:  
/rest/Spatial/erm/databases/dbsource/persistentUpdates/roadTypes/road_type.json  
  
or  
  
HTTP GET:  
/rest/Spatial/erm/databases/dbsource/persistentUpdates/roadTypes.json?roadTypes=road_type
```



ここで、**dbsource** は、永続更新を返すデータベースの名前で、**roadtype** は、更新を返す道路のタイプです。

注：最初の形式は、1つの道路タイプのみに対して永続更新を返すために使用します。2つめの形式は、複数またはすべての道路タイプに対して永続更新を返すために使用します。複数の道路タイプを指定する場合は、道路タイプをカンマで区切ります。すべての道路タイプを指定する場合は、パラメータ "roadtypes=" を空にします。以下の例を参照してください。

### HTTP DELETE URL 形式

HTTP DELETE 要求には次の形式を使用します。HTTP DELETE は、道路タイプに対する特定の永続更新を削除するために使用します。

```
HTTP DELETE:  
/rest/Spatial/erm/databases/dbsource/persistentUpdates/roadTypes/roadtype
```

ここで、**dbsource** は、データベースの名前で、**roadtype** は、削除する永続更新を含む道路のタイプです。Spatial モジュールのルーティングデータベースリソースツールで指定されたデータベース名を使用してください。

**roadtype** は、HTTP POST と HTTP DELETE の両方で次のいずれにでも設定できます。

- 進入路
- 裏道路
- 接続道路
- フェリー
- 小道
- 侵入制限道路 (密集都市部)
- 侵入制限道路 (農村部)
- 侵入制限道路 (郊外)
- 侵入制限道路 (都市部)
- 地方道路 (密集都市部)
- 地方道路 (農村部)
- 地方道路 (郊外)
- 地方道路 (都市部)
- 主要地方道路 (密集都市部)
- 主要地方道路 (農村部)
- 主要地方道路 (郊外)

- 主要地方道路 (都市部)
- 幹線道路 (密集都市部)
- 幹線道路 (農村部)
- 幹線道路 (郊外)
- 幹線道路 (都市部)
- 補助地方道路 (密集都市部)
- 補助地方道路 (農村部)
- 補助地方道路 (郊外)
- 補助地方道路 (都市部)
- 一般道路 (密集都市部)
- 一般道路 (農村部)
- 一般道路 (農村部)
- 一般道路 (都市部)
- 主要高速道路 (密集都市部)
- 主要高速道路 (農村部)
- 主要高速道路 (郊外)
- 主要高速道路 (都市部)
- 出入路 (密集都市部)
- 出入路 (侵入制限道路)
- 出入路 (幹線道路)
- 出入路 (主要高速道路)
- 出入路 (農村部)
- 出入路 (一般高速道路)
- 出入路 (都市部)
- 出入路 (郊外)
- 一般高速道路 (密集都市部)
- 一般高速道路 (農村部)
- 一般高速道路 (郊外)
- 一般高速道路 (都市部)

#### クエリ パラメータ

HTTP POST 操作は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
速度	文字列	いいえ	新しい速度を指定して道路タイプの新しい速度を定義する速度更新です。 <b>velocityUnit</b> パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
<b>velocityUnit</b>	文字列	いいえ	<b>velocity</b> または <b>velocityAdjustment</b> の速度の単位です。速度更新の場合、速度単位には mph (マイル/時) および kph (キロメートル/時) のいずれかの値を指定できます。デフォルト値は mph です。
<b>velocityAdjustment</b>	文字列	いいえ	速度 (単位と値) の変更を指定して道路タイプの速度の変化を定義する速度更新です。速度値は増加 (正の値) または減少 (負の値) させることができます。 <b>velocityUnit</b> パラメータを指定しない限り、デフォルトの単位は mph (マイル/時) です。
<b>velocityPercentage</b>	Integer	いいえ	速度を増加 (正の値) または減少 (負の値) させる割合を指定して道路タイプの速度の増加を定義する速度更新です。

## 例

### 道路タイプの新しい速度を設定する (HTTP POST)

```
http://<server>:<port>/rest/Spatial/em/databases/usrouteDatabase/persistentUpdates/roadTypes/ferry.json?velocity=5&velocityUnits=mph
```

### フェリーの道路タイプに対する更新のリストを返す (HTTP GET)

```
http://<server>:<port>/rest/Spatial/em/databases/US_NE/persistentUpdates/roadTypes/ferry.json?velocityUnit=kph
```

### US\_NE ルーティング データベース リソースのすべての道路タイプ更新のリストを返す (HTTP GET)

```
http://<server>:<port>/rest/Spatial/em/databases/US_NE/persistentUpdates/roadTypes.json?roadTypes=
```

フェリー、接続道路、一般道路 (都市部) の道路タイプに対する更新のリストを返す (HTTP GET)

```
http://<server>:<port>/rest/Spatial/erm/databases/US_NE/persistentUpdates/roadTypes.json?roadTypes=ferry,connector,normal  
road urban
```

道路タイプ永続更新を削除する (HTTP DELETE)

```
http://<server>:<port>/rest/Spatial/erm/databases/US_NE/persistentUpdates/roadTypes/back  
road
```

すべての更新を削除する

### HTTP DELETE URL 形式

HTTP DELETE 要求には次の形式を使用します。HTTP DELETE は、指定されたデータベースに対する永続更新をすべて削除するために使用します。

```
HTTP DELETE: /rest/Spatial/erm/databases/dbsource/persistentUpdates
```

ここで、**dbsource** は、削除する永続更新を含むデータベースの名前です。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

例

US\_NE ルーティング データベース リソースのすべての永続更新を削除します。

```
http://www.pb.com/rest/Spatial/erm/databases/US_NE/persistentUpdates
```

すべての更新の取得

### HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。この HTTP GET 操作は、指定されたルーティング データベース リソースのすべての永続更新を取得するために使用します。

```
HTTP GET: /rest/Spatial/erm/databases/dbsource/persistentUpdates.json
```

ここで、**dbsource** は、削除する永続更新を含むデータベースの名前です。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。

### クエリ パラメータ

この演算は、次のクエリ パラメータを取ります。

パラメータ	タイプ	必須	説明
<i>velocityUnit</i>	文字列	いいえ	サーバーに保存済みの更新が、この指定された単位で返ります。パラメータが指定されていない場合は、デフォルトの単位で応答が返ります。速度更新の場合、速度単位には <b>mph</b> (マイル/時)、 <b>kph</b> (キロメートル/時)、 <b>mtps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。デフォルト値は <b>mph</b> です。

### 例

US\_NE ルーティング データベース リソースの更新リストを返します。

```
http://<server>:<port>/rest/Spatial/erm/databases/US_NE/persistentUpdates.json
```

### 応答

```
{
  "roadTypeUpdates":
  [
    {
      "roadType": "major road dense urban",
      "speed":
      {
        "velocity": 90,
        "velocityUnit": "MPH"
      }
    }
  ],
  "segmentUpdates":
  [
    {
      "exclude": true,
      "roadType": "major road dense urban",
      "segmentID": "c75994cc:12d916",
      "speed":
```

```

    {
      "velocity": 65,
      "velocityUnit": "MPH"
    },
    {
      "exclude": true,
      "roadType": "major road dense urban",
      "segmentID": "7ac5401f:6b1bf7",
      "speed":
        {
          "velocity": 65,
          "velocityUnit": "MPH"
        }
    }
  ]
}

```

速度の単位パラメータが `kph` に指定されている場合。

```
http://<server>:<port>/rest/Spatial/erm/databases/<database_name>/persistentUpdates.json?velocityUnit=kph
```

応答

```

{
  "roadTypeUpdates": [{
    "roadType": "major road dense urban",
    "speed": {
      "velocity": 145,
      "velocityUnit": "KPH"
    }
  }]
}

```

## GetCapabilities

### 説明

GetCapabilities サービスにより、展開済みのルーティングエンジンに関するメタデータを取得できます。メタデータを使用して、サービスとその機能を確認することで、ルーティングサービスを使用する際のエクスペリエンスを最適化できます。

この機能は、REST サービスでのみ使用できます。

## HTTP GET URL 形式

HTTP GET 要求には次の形式を使用します。

```
http://<server>:<port>/rest/Spatial/erm/v1/capabilities.json
```

## クエリ パラメータ

次の表に、GetCapabilities のクエリ パラメータを示します。

パラメータ	必須	説明
acceptVersions	オプション	プレースホルダー (機能しません) を指定します。
sections	オプション 省略した場合は、すべてのセクションに関する情報を返します。	サービス メタデータ ドキュメントで返す対象にするセクションの名前が0個以上含まれる、カンマ区切りの順不同リストを指定します。セクション値の大文字と小文字は区別されません。使用可能なセクション値は次のとおりです。 <ul style="list-style-type: none"> <li>• ServiceIdentification</li> <li>• ServiceProvider</li> <li>• operationsMetadata</li> <li>• データベース</li> </ul>

## 応答

OGC GetCapabilities と同じ応答が返ります。JSON 形式で、次のセクションがあります。

- serviceIdentification
- serviceProvider
- operationsMetadata
- データベース

### serviceIdentification

このセクションには、この特定のサーバーに関する基本メタデータが含まれます。その内容の例を以下に示します。

```
"serviceIdentification":
  {
    "title": "PBS Routing Service",
    "abstract": "Routing service maintained by PBS",
    "keywords":
      {
        "keyword":
```

```

    [
    ],
    },
    "serviceType": "Routing",
    "serviceTypeVersion": "v1",
    "fees": "none",
    "accessConstraints": "none"
  }
}

```

この情報は、getCapabilities.json 設定ファイルで得られるものと同じです。

また、このファイルは *SpectrumDirectory\server\modules\routing* から入手できます。変更内容を有効にするには、サーバーを再起動する必要があります。管理者は、JSON ファイル内のエントリで、ユーザが、表示、変更、削除を行える情報を決定できます。JSON ファイルのフィールドは、すべて省略可能です。

### serviceProvider

このセクションには、このサーバーを運用する組織に関するメタデータが含まれます。その内容の例を以下に示します。

```

"serviceProvider":
{
  "providerName": "Routing Service Provider",
  "providerSite":
  {
    "href": "http://www.yourcompany.com/",
    "type": "simple"
  },
  "serviceContact":
  {
    "contactInfo":
    {
      "address":
      {
        "administrativeArea": "Province",
        "city": "City",
        "country": "Country",
        "deliveryPoint": "Mail Delivery Location",
"electronicMailaddress": "mailto://support@yourcompany.com",
        "postalCode": "PostCode"
      },
      "contactInstructions": "Contact Instructions",
      "hoursOfservice": "24 Hours",
      "phone":
      {
        "facsimile": "1.800.000.0000",
        "voice": "1.800.000.0000"
      }
    }
  },
}

```



```

        "individualName": "Contact Person",
        "positionName": "Contact Person's Title",
        "role": "Contact Person's Role"
    }
}

```

このセクションも、上記の `getCapabilities.json` 設定ファイルを用いて設定されます。

### operationsMetadata

このセクションには、操作を要求するための URL など、このサーバーによって実装されている操作に関するメタデータが含まれます。あらかじめ定められている次の操作またはサービスが、このセクションに記述されます。

- **GetRoute**: ポイント間サービス
- **GetRouteCostMatrix**: ポイント処理サービスのマトリクス
- **GetTravelBoundary**: 自動車または徒歩による移動時間、または距離の限界を生成します。
- **DescribeDatasets**: 設定済みデータセットに関する情報を提供
- **DescribeDatabases**: 設定済みのすべてのデータベースに関する情報を提供します。
- **GetSegmentDataForPoint**: ポイントのセグメント情報を返します。
- **GetSegmentDataForSegment**: セグメント ID のセグメント情報を返します。
- **ListPersistentUpdates**: サーバーに存在するすべての Persistent Update をリスト表示します。
- **DeletePersistentUpdates**: サーバーに存在する、すべての Persistent Update を削除します。
- **SetPersistentUpdatesAtPoint**: サーバーにある、指定したポイントの Persistent Update を保存します。
- **SetPersistentUpdatesForSegment**: サーバーにある、指定したセグメント ID の Persistent Update を保存します。
- **SetPersistentUpdatesForRoadType**: サーバーにある、指定した道路タイプの Persistent Update を保存します。

その内容の例を以下に示します。

```

{"operationsMetadata": [{
  "name": "GetRoute",
  "DCP": {
    "HTTP": {
      "GET":
"<schema>://<server>:<port>/rest/Spatial/em/databases/<DB_NAME>.json?q=route&version=2",

      "POST":
"<schema>://<server>:<port>/rest/Spatial/em/databases/<DB_NAME>.json?q=route&version=2"
    }
  },
  "parameter": {
    "name": "OutputFormat",

```

```

    "value": "text/json"
  }
},

{
  "name": "GetRouteCostMatrix",
  "DCP": {
    "HTTP": {
      "GET":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>.json?q=routeCostMatrix&version=2",

      "POST":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>.json?q=routeCostMatrix&version=2"

    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"}
},

{
  "name": "GetTravelBoundary",
  "DCP": {
    "HTTP": {
      "GET":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>.json?q=travelBoundary&version=2",

      "POST":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>.json?q=travelBoundary&version=2"

    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

{
  "name": "DescribeDatasets",
  "DCP": {
    "HTTP": {
      "GET": "<schema>://<server>:<port>/rest/Spatial/erm/v1/datasets.json"

    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},
},

```

```

{
  "name": "DescribeDatabases",
  "DCP": {
    "HTTP": {
      "GET":
"<schema>://<server>:<port>/rest/Spatial/erm/v1/databases.json"
    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

{
  "name": "GetSegmentDataForPoint",
  "DCP": {
    "HTTP": {
      "GET":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>/segments.json"
    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

{
  "name": "GetSegmentDataForSegment",
  "DCP": {
    "HTTP": {
      "GET":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>/segments/<segmentID>.json"
    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

{
  "name": "ListPersistentUpdates",
  "DCP": {
    "HTTP": {
      "GET":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>/persistentUpdates.json"
    }
  }
}

```

```

    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

{
  "name": "DeletePersistentUpdates",
  "DCP": {
    "HTTP": {
      "DELETE":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>/persistentUpdates"

    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

{
  "name": "SetPersistentUpdatesAtPoint",
  "DCP": {
    "HTTP": {
      "POST":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>/persistentUpdates.json"

    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

{
  "name": "SetPersistentUpdatesForSegment",
  "DCP": {
    "HTTP": {
      "POST":
"<schema>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>/persistentUpdates/segments/<segmentID>.json"

    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
},

```

```

{
  "name": "SetPersistentUpdatesForRoadType",
  "DCP": {
    "HTTP": {
      "POST":
"<schem>://<server>:<port>/rest/Spatial/erm/databases/<DB_NAME>/persistentUpdates/roadTypes/<roadtype>.json"
    }
  },
  "parameter": {
    "name": "OutputFormat",
    "value": "text/json"
  }
}
]
}

```

## データベース

このセクションには、サーバーで設定済みのデータベースの名前のリストが含まれます。

例:

```

{
  "databases":
  [
    "US_NE",
    "US"
  ]
}

```

データベースがサーバー上に設定されていない場合は、以下のものが返されます。

```

{
  "databases": [
  ]
}

```

## DescribeDatasets

### 説明

DescribeDatasets サービスでは、Spectrum™ Technology Platform サーバーに追加されたルーティング データベースに対応するデータセットに関するメタデータ情報を取得できます。応答は、データセット パスに存在するメタデータ情報に似たものになります。

この機能は、REST サービスでのみ使用できます。

## HTTP GET URL 形式

**すべてのデータセット:** すべてのデータセットを対象にした HTTP GET リクエストには、次の形式を使用します。

```
http://<server>:<port>/rest/Spatial/erm/v1/datasets.json
```

**単一のデータセット:** 単一のデータセットを対象にした HTTP GET リクエストには、次の形式を使用します。

```
http://<server>:<port>/rest/Spatial/erm/v1/datasets/<dataset_ID>.json
```

**dataset\_ID** は、DescribeDatabases サービスからの 'dataSets' 配列の中の要素に対応する 'id' です。

## 応答

このサービスの応答は、JSON 配列です。

すべてのデータセットの場合、JSON 配列の長さは、Management Console で設定したデータベースに追加された、(メタデータが存在する)データセットパスの総数と同じです。データセットパスにメタデータが存在しない場合、そのエントリは無視されます。

1つのデータセットの場合、JSON 配列の長さは、データセットパスにメタデータが存在する場合に限り、1 となります。それ以外の場合は、空の JSON 配列が返ります。

## 例

2つのルーティングデータベースが Spectrum™ Technology Platform に追加されているとします。データベースの名前とデータセットパスは以下のとおりです。

1. **US\_NE:** E:\\db\\ERM-US\\2014.09\\driving\\northeast
2. **US:** E:\\db\\ERM-US\\2014.09\\driving\\midwest と  
E:\\db\\ERM-US\\2014.09\\driving\\south

**サンプル リクエスト (すべてのデータセット):**

```
http://<server>:<port>/rest/Spatial/erm/v1/datasets.json
```

**レスポンス:**

```
{
  "dataSets": [{
    "component": "routing",
    "description": "USA Test dataset",
    "ext": {
      "bbox": [68.291015625, 7.9721977144, 97.55859375, 35.4606699515],
      "crs": "epsg:4326",
```

```

    "cvr": true,
    "historicTrafficTimeBuckets": {
      "amPeak": {
        "lowerBound": 700,
        "upperBound": 1000
      },
      "nightTime": {
        "lowerBound": 2200,
        "upperBound": 400
      },
      "offPeak": {
        "lowerBound": 1000,
        "upperBound": 1600
      },
      "pmPeak": {
        "lowerBound": 1600,
        "upperBound": 1900
      }
    },
    "locale": "EN",
    "type": "driving"
  },
  "id": "US dataset",
  "name": "USA",
  "product": "Spatial",
  "vintage": "September 2015"
}
}
}

```

### サンプル リクエスト (1つのデータセット):

```
http://<server>:<port>/rest/Spatial/erm/v1/datasets/US%20dataset.json
```

### レスポンス:

```

{
  "dataSets": [{
    "component": "routing",
    "description": "USA Test dataset",
    "ext": {
      "bbox": [68.291015625, 7.9721977144, 97.55859375, 35.4606699515],

      "crs": "epsg:4326",
      "cvr": true,
      "historicTrafficTimeBuckets": {
        "amPeak": {
          "lowerBound": 700,
          "upperBound": 1000
        },
        "nightTime": {
          "lowerBound": 2200,
          "upperBound": 400
        }
      }
    }
  }
]
}

```

```

    },
    "offPeak": {
      "lowerBound": 1000,
      "upperBound": 1600
    },
    "pmPeak": {
      "lowerBound": 1600,
      "upperBound": 1900
    }
  },
  "locale": "EN",
  "type": "driving"
},
"id": "US dataset",
"name": "USA",
"product": "Spatial",
"vintage": "September 2015"
}]
}

```

## DescribeDatabases

### 説明

DescribeDatabases 操作は、システムで設定済みで、リクエストで使用可能な、すべてのデータベース リソースの名前を返します。システム内のすべてのデータベースの名前を含むリストと、各データベースのデータセットを含む配列が返されます。

### HTTP GET URL 形式 (すべてのデータベース)

HTTP GET リクエストには次の形式を使用します。データ リソースがサーバー上に存在しない場合は、空のリストが返されます。

```
http://<server>:<port>/rest/Spatial/erm/v1/databases.json
```

### 例 (すべてのデータベース)

リクエスト:

```
http://<server>:<port>/rest/Spatial/erm/v1/databases.json
```

レスポンス:

```

{
  "databases":
  [
    {
      "dataSets":

```



```

    [
      "US_Central"
    ],
    "name": "US_CN"
  },
  {
    "dataSets":
    [
      "US_NorthEast"
    ],
    "name": "US_NE"
  },
  {
    "dataSets":
    [
      "US_Central",
      "US_Midwest",
      "US_NorthEast",
      "US_Pacific",
      "US_South"
    ],
    "name": "US"
  }
]
}

```

### HTTP GET URL 形式 (1 つのデータベース)

HTTP GET リクエストには次の形式を使用します。特定のデータリソースに関するデータセット情報を取得する場合に、このリクエストを使用します。指定された名前のデータリソースがサーバー上に存在しない場合は、例外が返されます。

```
http://<server>:<port>/rest/Spatial/erm/v1/<database_name>.json
```

### 例 (1 つのデータベース)

リクエスト:

```
http://<server>:<port>/rest/Spatial/erm/v1/databases/US.json
```

レスポンス:

```

{
  "databases":
  [
    {
      "dataSets":
      [
        "US_Central",
        "US_Midwest",

```

```

    "US_NorthEast",
    "US_Pacific",
    "US_South"
  ],
  "name": "US"
}
]
}

```

## Universal Addressing モジュール

### AutoCompleteLoqate

**AutoCompleteLoqate** は、住所データのリアルタイム入力に対して高速で正確な結果を返します。フォームに入力された各文字に基づいて瞬時に結果が返され、正確なデータのみがデータベースに入力されることを保証します。**AutoCompleteLoqate** には [高度な検索を優先] オプションも用意されており、インデックス ファイル形式のデータを使用することによって、238 カ国に対して入力時間を最大 80% 短縮します。

#### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

#### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

**注:** この例では、内容を理解しやすいように空の応答要素を削除し、最初の 3 つの住所マッチのみを示しています。

```

{"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": []
  }
}
]]

```

### XML 応答の例

次の例では、XML 応答を要求します。

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

この要求から次のような XML が返されます。

**注:** この例では、内容を理解しやすいように空の応答要素を削除し、最初の3つの住所マッチのみを示しています。

```

<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>

```

## 要求

### 入力データのパラメータ

AutoCompleteLoqate の入力の一覧を以下の表に示します。

**表 16:** 入力フォーマット

パラメータ	説明
Data.AddressLine1	最初の住所行。
Data.AddressLine2	2 行目の住所行。

パラメータ	説明
Data.AddressLine3	3 行目の住所行。
Data.AddressLine4	4 行目の住所行。
Data.City	都市名。
Data.Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2 桁の ISO 国コード</li> <li>• 3 桁の UPU 国コード</li> <li>• 英語の国名</li> </ul> <p>ISO コードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a>（735ページ）を参照してください。</p>
Data.FirmName	会社名または企業名。
Data.PostalCode	住所の郵便番号。
Data.StateProvince	州または省。

### オプションのパラメータ

表 17 : **AutoCompleteLoqate** のオプション

パラメータ	説明
Option.Database.Loqate	住所処理に使用するデータベースを指定します。Management Console の [データベース リソース] パネルで定義されたデータベースのみが使用可能です。

## パラメータ

## 説明

## Option.OutputCasing

出力データの大文字と小文字の区別を指定します。次のいずれかです。

**M** 出力に大文字と小文字を混在させます (デフォルト)。例:

123 Main StMytown FL 12345

**U** 出力に大文字を使用します。例:

123 MAIN STMYTOWN FL 12345

## Option.HomeCountry

デフォルトの国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がドイツにある場合は、ドイツを指定します。有効な国名には次のものがあります。

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

パラメータ	説明
OutputCountryFormat	<p><b>Country</b> 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。</p> <p><b>E</b> 英語の国名を使用します (デフォルト)。</p> <p><b>I</b> 国名の代わりにその国の 2 文字の ISO の略語を使用します。</p> <p><b>U</b> 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。</p>
Option.OutputScript	<p>出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。</p> <p><b>Input</b> 書き直しを実行せず、入力と同じスクリプトで出力します (デフォルト)。</p> <p><b>Native</b> 使用可能な場合は、選択した国のネイティブ スクリプトで出力します。</p> <p><b>Latn</b> 英語の値を使用します。</p>
Option.MaximumResults	AutoCompleteLoqate が返す住所の最大数。デフォルトは 10 です。

---

パラメータ	説明
-------	----

---

Option.isPowersearchEnable	インデックス ファイル形式のデータを使用することによって、240 カ国に対して入力時間を最大 80% 短縮します。検索を実行すると、Loqate エンジンはず、該当するインデックスを検索します。インデックスが存在する場合は、候補住所のリストを直ちに返そうと試みます。インデックスが存在しないか、インデックスによって結果が1つも返されない場合は、オリジナルの検索処理を開始します。
----------------------------	---

注：高度な検索は、入力ファイルにフィールドが 2 つだけ存在する場合に実行可能です。1 つは **Country** フィールドで、もう 1 つはいずれかの **AddressLine** フィールドです。このオプションを選択し、入力ファイルにそれ以外のフィールドが含まれる場合は、オリジナルの検索処理が自動的に開始されます。

検索を行うために、**Auto Complete** インデックスは米国内の検索に対しては最初の 10 文字まで、その他すべての対象国内の検索に対しては最初の 15 文字までを使用します。空白と句読文字は、この文字数にカウントされません。

高度な検索は、ボツワナ、エチオピア、インド、カザフスタン、マレーシア、モンゴル、セントクリストファー・ネイビス、およびサンマリノでは使用できません。

注：高度な検索を使用するには、有効なライセンスが必要です。高度な検索のライセンスを取得していないか、ライセンスの期限が切れている状態でこのオプションを選択すると、エラーとなります。

---

Option.IsDuplicateHandlingMaskEnable	重複処理マスクを有効にし、重複レコードの処理および削除の方法を指定します。次のオプションから 1 つ以上を選択します。
--------------------------------------	---

- S** デフォルトで選択されています。入力の前処理により、単一フィールドで発生している重複を削除します。
  - C** デフォルトで選択されています。入力の前処理により、すべてのフィールドわたり重複を削除します。
  - T** 入力の前処理により、標準住所フィールドでないフィールド内の重複を削除します。
  - F** デフォルトで選択されています。検証の出力の後処理により、検証されていないフィールドから重複を削除します。
-



パラメータ	説明
Option.FailJobOnDataLicenseError	データ ライセンス エラーの発生時に Spectrum Technology Platform がどのように応答するかを指定します。
ジョブのエラー	データライセンスエラーが発生した場合、ジョブ全体をエラーにします。
レコードのエラー	データライセンスエラーの発生原因となったレコードをエラーにし、処理を続行します。

## 応答

AutoCompleteLoqate の出力はオプションであり、[AutoCompleteLoqate オプション] ダイアログボックスの [出力フィールド] セクションで選択したフィールドに直接対応します。

表 18 : AutoCompleteLoqate の出力

応答要素	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。
AddressLine4	4 行目の住所行。
City	都市名。
Country	3 文字の ISO 3166-1 Alpha 3 国コード。ISO コードの一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。
FirmName	企業名。

応答要素	説明
HouseNumber	候補住所の家番号が含まれる範囲の終了家番号。
PostalCode	郵便番号。
PostalCode.AddOn	ZIP + 4 <sup>®</sup> Code の末尾 4 桁。
ProcessedBy	住所を処理した住所コーダーを示します。 <b>LOQATE</b> Loqate コーダーが住所を処理しました。
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。 <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• RequestFailed</li> <li>• NoLookupAddressFound</li> </ul>
Status.Description	問題の説明 (ある場合)。 <b>Did not return multiples</b> 入力住所はデータベース内の 1 つの住所とのみマッチしました。AutoCompleteLoqate は、一致する可能性のある住所が複数見つかった場合のみデータを返します。 <b>Not able to look up the address pattern</b> AutoCompleteLoqate は、部分的な住所を処理できません。

### AutoCompleteLoqate のサンプル Web アプリケーション

Auto Complete Loqate 機能を示すサンプル Web アプリケーションにアクセスできます。部分的な住所を入力すると、このアプリケーションは Auto Complete Loqate REST Web サービスを呼び出します。この Web サービスは提案される住所を返します。

注：この機能を使用する前に、Management Console で Auto Complete Loqate データベースリソースを追加し、そのデータベースリソースを Auto Complete Loqate Service に保存する必要があります。

1. Spectrum™ Technology Platform サーバーが実行されていることを確認します。
2. Web ブラウザを開いて、`http://<servername>:<port>/autocomplete` に移動します。たとえば、サーバーの名前が "myserver" であり、既定の HTTP ポート 8080 を使う場合は、`http://myserver:8080/autocomplete` に移動します。

注：このサイトは、Internet Explorer 8.0 またはそれ以降、Chrome、Mozilla Firefox で最も適切に表示されます。

3. ログイン画面が表示されたら、ユーザ名として "guest" を入力し、パスワードフィールドは空白のままにします。
4. [OK] をクリックします。
5. ドロップダウンリストから国を選択します。
6. 提供されている任意のフィールドで住所の入力を開始します。
7. 提案された住所のリストから選択します。
8. 新しい呼び出しを開始するには、[リセット] をクリックして、前回の呼び出しで使用したフィールドをクリアします。

### GetCandidateAddresses

GetCandidateAddresses は、与えられた入力住所にマッチするとみなされる住所のリストを返します。GetCandidateAddresses は、入力住所が郵便データベースの複数の住所にマッチする場合のみ、候補の住所を返します。入力住所が、郵便データベースの 1 つの住所のみにマッチする場合は、住所データを返しません。

米国およびカナダ以外の住所については、ValidateAddress が返す複数のマッチ結果と、GetCandidateAddresses が同じ住所に対して返す結果の間に、矛盾が存在する場合があります。矛盾した結果が得られるのはおそらく、ValidateAddress のパフォーマンス チューニング設定で 100 以外の値を設定しているためです。GetCandidateAddresses と ValidateAddress で矛盾のない結果を得るには、パフォーマンス チューニング オプションを 100 に設定します。

注：デフォルトでは、GetCandidateAddresses は個々の家番号との一致は確認しません。各通りの家番号の範囲とのマッチングを行います。GetCandidateAddresses は通り名、都市名、州/省名、および郵便番号を特定した後、入力された家番号が、マッチした通り名の

家番号の範囲に含まれるかどうかを確認します。ユニット番号についても同様の処理が行われます。個々の家番号が有効であることを確認するには、**ValidateAddress Delivery Point Validation (DPV)** 処理オプションを使用する必要があります。DPV 処理は、米国住所に対してのみ使用可能です。

カナダのコーダーには、特定の郵便番号を入力として受け取り、その郵便番号のデータベースに格納された通り情報を返す逆検索ルーチンが含まれています。この機能を使用するには、**PostalCode** フィールドにカナダの郵便番号を入力します。カナダの郵便番号を入力した場合の結果については、以下の 2 つめの例を参照してください。

**GetCandidateAddresses** は、**Universal Addressing** モジュールに含まれています。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

```
{ "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": []
  }
]}

```

### XML 応答の例

次の例では、XML 応答を要求します。

```

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

```

この要求から次のような XML が返されます。

```
<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>
```

## 要求

入力データのパラメータ

GetCandidateAddresses の入力の一覧を以下の表に示します。

表 19 : 入力フォーマット

パラメータ	説明
Data.AddressLine1	最初の住所行。
Data.AddressLine2	2 行目の住所行。
Data.AddressLine3	3 行目の住所行。 米国またはカナダの住所には適用されません。
Data.AddressLine4	4 行目の住所行。 米国またはカナダの住所には適用されません。
Data.AddressLine5	5 行目の住所行。 英国住所にのみ適用されます。通り名、ユニット番号、ビルディング番号などを含めることができます。
Data.City	都市名。
Data.StateProvince	州または省。 米国の住所に限り、州を StateProvince フィールドの代わりに City フィールドに入れることができます。
Data.PostalCode	住所の郵便番号。米国住所の場合は、次のいずれかの形式の ZIP Code™ になります。  9999999999-9999A9A9A9A9A 9A99999 999  注：カナダ住所の場合は、このフィールドのみを入力すると、候補となる住所データが返されます。その他の国の場合は、AddressLine1 と AddressLine2 も入力する必要があります。

パラメータ	説明
Data.Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"><li>• 2 桁の ISO 国コード</li><li>• 3 桁の UPU 国コード</li><li>• 英語の国名</li><li>• フランス語の国名</li><li>• ドイツ語の国名</li><li>• スペイン語の国名</li></ul> <p>ISO コードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a> (735ページ) を参照してください。</p>
Data.FirmName	会社名または企業名。
Data.USUrbanName	米国住所都市化名。主にプエルトリコの住所に使用します。



## オプションのパラメータ

表 20 : GetCandidateAddresses オプション

パラメータ	説明
Option.PerformUSProcessing	<p>米国住所を処理するかどうかを指定します。米国住所処理を有効にすると、GetCandidateAddresses は、米国住所の候補となる住所の取得を試みます。米国住所処理を無効にすると、米国住所は失敗します。つまり、Status 出力フィールドに "F" が設定されて返されます。出力フィールド Status.Code は、"DisabledCoder" となります。米国住所処理のライセンスを取得していない場合は、ジョブに米国住所が含まれるか否かにかかわらず、米国住所処理を無効にしなければ、ジョブを正常に実行することはできません。</p> <p>注：米国住所を正常に処理するには、米国住所処理の有効なライセンスを取得する必要があります。米国住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、米国住所処理を有効にすると、エラーとなります。</p> <p><b>Y</b>      はい、米国住所を処理します (デフォルト)。</p> <p><b>N</b>      いいえ、米国住所を処理しません。</p>
Option.Database.US	<p>米国住所処理に使用するデータベースを指定します。Management Console の <b>【米国データベース リソース】</b> パネルで定義されたデータベースのみが使用可能です。</p>

## パラメータ

## 説明

---

**Option.PerformCanadianProcessing**

カナダ住所を処理するかどうかを指定します。カナダ住所処理を有効にすると、**GetCandidateAddresses** は、カナダ住所の候補となる住所の取得を試みます。カナダ住所処理を無効にした場合、**Status** フィールドに "F" が設定されカナダ住所は失敗します。出力フィールド **Status.Code** は、"DisabledCoder" となります。カナダ住所処理のライセンスを取得していない場合は、ジョブにカナダ住所が含まれるか否かにかかわらず、カナダ住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：カナダ住所を正常に処理するには、カナダ住所処理の有効なライセンスを取得する必要があります。カナダ住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、カナダ住所処理を有効にすると、エラーとなります。

**Y**      カナダ住所を処理します (デフォルト)。

**N**      カナダ住所を処理しません。

---

**Option.Database.Canada**

カナダ住所処理に使用するデータベースを指定します。**Management Console** の **[カナダ データベース リソース]** パネルで定義されたデータベースのみが使用可能です。

---

**Option.PerformInternationalProcessing**

国際住所 (米国およびカナダ以外の住所) を処理するかどうかを指定します。国際住所処理を有効にすると、**GetCandidateAddresses** は、国際住所の候補となる住所の取得を試みます。国際住所処理を無効にした場合、**Status** フィールドに "F" が設定され国際住所は失敗します。出力フィールド **Status.Code** は、"DisabledCoder" となります。国際住所処理のライセンスを取得していない場合は、ジョブに国際住所が含まれるか否かにかかわらず、国際住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：国際住所を正常に処理するには、国際住所処理の有効なライセンスを取得する必要があります。国際住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、国際住所処理を有効にすると、エラーとなります。

**Y**      国際住所を処理します (デフォルト)。

**N**      国際住所を処理しません。

パラメータ	説明
Option.Database.International	国際的な住所の検証に使用するデータベースを指定します。 Management Console の【国際データベース リソース】パネルで定義されたデータベースのみが使用可能です。
Option.OutputCasing	出力データの大文字と小文字の区別を指定します。次のいずれかです。 <b>M</b> 出力に大文字と小文字を混在させます (デフォルト)。例: 123 Main StMytown FL 12345 <b>U</b> 出力に大文字を使用します。例: 123 MAIN STMYTOWN FL 12345
Option.MaximumResults	GetCandidateAddresses が返す候補住所の最大数。デフォルトは 10 です。最大数は 10 になります。
Option.OutputShortCityName	米国住所に対しては、USPS® が承認する都市の略称がある場合に、それを返すかどうかを指定します。USPS® は、14 文字以上の都市名に対し、略称を定めています。都市の略称は 13 文字以下で、宛名ラベルのサイズが限られている場合に使用できます。短い都市名が存在しない都市に対しては、正式な都市名が返されます。 <b>Y</b> 短い都市名を返します。 <b>N</b> 短い都市名を返しません。

## パラメータ

## 説明

## Option.DualAddressLogic

(米国住所のみ)。ストリート情報と PO Box/地方配送路/Highway Contract 情報の両方が住所に含まれる場合に、GetCandidateAddresses が、ストリート一致を返すか、または PO Box/地方配送路/Highway Contract 一致を返すかを制御します。詳細については、[二重住所ロジックについて](#) (588ページ) を参照してください。

- N** (デフォルト)USPS®CASS™ の規則では、以下の優先順位に基づいて、返す住所を決定します。
1. PO Box
  2. Firm
  3. Highrise
  4. Street
  5. Rural Route
  6. General Delivery
- S** 住所行に関係なく、通り一致を返します。
- P** 住所行に関係なく、PO Box 一致を返します。

## Option.StreetMatchingStrictness

通り名のマッチングの精度 (米国住所のみ)。

- E** 入力された通り名は、データベースに完全に一致する必要があります。
- T** マッチングアルゴリズムは "厳格" です。
- M** マッチングアルゴリズムは "中" です (デフォルト)。
- L** マッチングアルゴリズムは "あいまい" です。

## Option.FirmMatchingStrictness

企業名マッチングの精度 (米国住所のみ)。

- E** 入力された企業名は、データベースに完全に一致する必要があります。
- T** マッチングアルゴリズムは "厳格" です。
- M** マッチングアルゴリズムは "中" です (デフォルト)。
- L** マッチングアルゴリズムは "あいまい" です。

## パラメータ

## 説明

Option.DirectionalMatchingStrictness	<p>道順マッチングの精度。</p> <p><b>E</b> 入力された道順は、データベースに完全に一致する必要があります。</p> <p><b>T</b> マッチングアルゴリズムは "厳格" です。</p> <p><b>M</b> マッチングアルゴリズムは "中" です (デフォルト)。</p> <p><b>L</b> マッチングアルゴリズムは "あいまい" です。</p>
Option.PerformESM	<p>Enhanced Street Matching (ESM) を実行するかどうかを指定します。ESM は、通常の住所検証プロセスでマッチしなかった任意の入力住所に対して、追加データによる別のマッチングロジックを適用します。ESM は、米国住所にのみ適用されます。</p> <p><b>Y</b> ESM 処理を実行します。</p> <p><b>N</b> ESM 処理を実行しません (デフォルト)。</p>
Option.AddressLineSearchOnFail	<p>ValidateAddress において、住所行で都市、州/省、郵便番号の検索を行うかどうかを指定します。</p> <p>このオプションにより、ValidateAddress において、City、StateProvince、および PostalCode の各入力フィールドの値を使用して住所にマッチする結果が得られなかった場合に、AddressLine 入力フィールドで都市、州/省、郵便番号、および国を検索することができます。</p> <p>入力住所において、AddressLine フィールドに都市、州/省、および郵便番号の情報が存在する場合は、このオプションを有効にすることを検討してください。</p> <p>入力住所において、City、State/Province、および PostalCode フィールドが使用されている場合は、このオプションを無効にしてください。このオプションを有効にしてこれらのフィールドを使用すると、ValidateAddress がこれらのフィールド値の修正 (例えば、スペルミスのある都市名など) に失敗する可能性が高くなります。</p> <p><b>Y</b> AddressLine フィールドを検索します (デフォルト)。</p> <p><b>N</b> いいえ、AddressLine フィールドを検索しません。</p>

## 応答

GetCandidateAddresses は、次の出力を返します。

表 21 : GetCandidateAddresses の出力

応答要素	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。
AddressLine4	4 行目の住所行。
AddressLine5	英国住所にのみ適用されます。住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 5 行目です。住所の妥当性が確認できなかった場合は、入力住所の 5 行目がそのまま出力されます。
CanadianDeliveryInstallation AreaName	配達施設名 (カナダ住所のみ)
CanadianDeliveryInstallation QualifierName	配達施設の修飾子 (カナダ住所のみ)
CanadianDeliveryInstallation Type	配達施設の種類 (カナダ住所のみ)
City	都市名。
Country	3 文字の ISO 3166-1 Alpha 3 国コード。ISO コードの一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。
FirmName	企業名。

応答要素	説明
HouseNumberHigh	候補住所の家番号が含まれる範囲の終了家番号。
HouseNumberLow	候補住所の家番号が含まれる範囲の開始家番号。
HouseNumberParity	HouseNumberLow と HouseNumberHigh の間の家番号の番号付けスキームを次のように示します。 <ul style="list-style-type: none"> <li><b>E</b> 偶数値のみ</li> <li><b>O</b> 奇数値のみ</li> <li><b>B</b> 両方</li> </ul>
MatchLevel	米国およびカナダ以外の住所に対し、候補住所のマッチ レベルを示します。米国およびカナダの住所では常に "A" になります。次のいずれかです。 <ul style="list-style-type: none"> <li><b>A</b> 候補は通りレベルで入力住所にマッチします。</li> <li><b>B</b> 候補は州/省レベルで入力住所にマッチします。</li> </ul>
PostalCode	郵便番号。米国では、ZIP Code™になります。
PostalCode.AddOn	ZIP + 4® Code の末尾 4 桁。米国住所にのみ適用されます。
RecordType	米国およびカナダの郵政当局によって定義されている住所レコードのタイプ(米国およびカナダの住所のみサポート): <ul style="list-style-type: none"> <li>• FirmRecord</li> <li>• GeneralDelivery</li> <li>• HighRise</li> <li>• PostOfficeBox</li> <li>• RRHighwayContract</li> <li>• Normal</li> </ul>
RecordType.Default	"デフォルト" マッチを示すコード <ul style="list-style-type: none"> <li><b>Y</b> 住所はデフォルト レコードにマッチしています。</li> <li><b>NULL</b> 住所はデフォルト レコードにマッチしていません。</li> </ul>

応答要素	説明										
StateProvince	州または省の省略形。										
Status	<p>マッチの成功または失敗。</p> <table> <tr> <td><b>NULL</b></td> <td>成功</td> </tr> <tr> <td><b>F</b></td> <td>失敗</td> </tr> </table>	<b>NULL</b>	成功	<b>F</b>	失敗						
<b>NULL</b>	成功										
<b>F</b>	失敗										
Status.Code	<p>失敗の原因 (ある場合)。次のいずれかの値になります。</p> <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• RequestFailed</li> </ul>										
Status.Description	<p>問題の説明 (ある場合)。</p> <table> <tr> <td><b>Did not return multiples</b></td> <td>入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddresses は、一致する可能性のある住所が複数見つかった場合にデータを返します。</td> </tr> <tr> <td><b>Number of candidates is not greater than 1</b></td> <td>入力住所はデータベース内の複数の住所にマッチしましたが、住所が返されませんでした。</td> </tr> <tr> <td><b>PerformUSProcessing disabled</b></td> <td>Status.Code=DisabledCoder の場合にこの値が表示されます。</td> </tr> <tr> <td><b>PerformCanadianProcessing disabled</b></td> <td>Status.Code=DisabledCoder の場合にこの値が表示されます。</td> </tr> <tr> <td><b>PerformInternationalProcessing disabled</b></td> <td>Status.Code=DisabledCoder の場合にこの値が表示されます。</td> </tr> </table>	<b>Did not return multiples</b>	入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddresses は、一致する可能性のある住所が複数見つかった場合にデータを返します。	<b>Number of candidates is not greater than 1</b>	入力住所はデータベース内の複数の住所にマッチしましたが、住所が返されませんでした。	<b>PerformUSProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。	<b>PerformCanadianProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。	<b>PerformInternationalProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。
<b>Did not return multiples</b>	入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddresses は、一致する可能性のある住所が複数見つかった場合にデータを返します。										
<b>Number of candidates is not greater than 1</b>	入力住所はデータベース内の複数の住所にマッチしましたが、住所が返されませんでした。										
<b>PerformUSProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。										
<b>PerformCanadianProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。										
<b>PerformInternationalProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。										
UnitNumberHigh	候補住所のユニット番号が含まれる範囲の終了ユニット番号。										
UnitNumberLow	候補住所のユニット番号が含まれる範囲の開始ユニット番号。										



応答要素	説明
UnitNumberParity	UnitNumberLow と UnitNumberHigh の間のユニット番号の番号付けスキームを次のように示します。 <b>E</b> 偶数値のみ <b>O</b> 奇数値のみ <b>B</b> 両方
USUrbanName	妥当性が確認された都市の都市化名。都市化名は、主にプエルトリコ住所に使用されます。

### GetCandidateAddressesLoqate

GetCandidateAddressesLoqate は、与えられた入力住所に一致するとみなされる住所のリストを返します。GetCandidateAddressesLoqate は、入力住所が郵便データベースの複数の住所にマッチする場合のみ、候補の住所を返します。入力住所が、郵便データベースの1つの住所のみにマッチする場合は、住所データを返しません。[Country] 入力フィールドは必須です。このフィールドが空白の場合、出力は返されません。

注：デフォルトでは、GetCandidateAddressesLoqate は個々の家番号との一致は確認しません。各通りの家番号の範囲とのマッチングを行います。GetCandidateAddressesLoqate は通り名、都市名、州/省名、および郵便番号を特定した後、入力された家番号が、マッチした通り名の家番号の範囲に含まれるかどうかを確認します。ユニット番号についても同様の処理が行われます。

GetCandidateAddressesLoqate は、Universal Addressing モジュールに含まれています。

#### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

## JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

注：この例では、空の応答要素を削除しています。最初の候補住所を 2 つだけ示します。

```
{ "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": []
  }
] }
```

## XML 応答の例

次の例では、XML 応答を要求します。

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

この要求から次のような XML が返されます。

注：この例では、空の応答要素を削除しています。最初の候補住所を 2 つだけ示します。

```
<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>

```

**要求**

## 入力データのパラメータ

GetCandidateAddressesLoqate の入力の一覧を以下の表に示します。

**表 22 : 入力フォーマット**

パラメータ	説明
Data.AddressLine1	最初の住所行。
Data.AddressLine2	2 行目の住所行。
Data.AddressLine3	3 行目の住所行。
Data.AddressLine4	4 行目の住所行。
Data.City	都市名。

パラメータ	説明
Data.Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2 桁の ISO 国コード</li> <li>• 3 桁の UPU 国コード</li> <li>• 英語の国名</li> </ul> <p>ISO コードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a> (735ページ) を参照してください。</p> <p>注：このフィールドは必須です。このフィールドが空白の場合、出力は返されません。</p>
Data.FirmName	会社名または企業名。
Data.PostalCode	住所の郵便番号。米国住所の場合は、次のいずれかの形式の ZIP Code™ になります。
Data.StateProvince	<p>州または省。</p> <p>米国住所に限り、州は、StateProvince フィールドではなく、City フィールドに入力することもできます。</p>

## オプションのパラメータ

表 23 : GetCandidateAddressesLoqate のオプション

パラメータ	説明
Option.Database.Loqate	住所処理に使用するデータベースを指定します。Management Console で定義されたデータベースのみが使用可能です。

パラメータ	説明
Option.OutputCasing	<p>出力データの大文字と小文字の区別を指定します。次のいずれかです。</p> <p><b>M</b> 出力に大文字と小文字を混在させます (デフォルト)。例: 123 Main StMytown FL 12345</p> <p><b>U</b> 出力に大文字を使用します。例: 123 MAIN STMYTOWN FL 12345</p>
Option.CandidateProcessOption	<p>候補を検索する方法を指定します。次のいずれかです。</p> <p><b>S</b> 住所の全体または一部を入力し、近似一致結果のリストを出力として返します (デフォルト)。</p> <p><b>V</b> 住所行、住所コンポーネント、またはその両方を組み合わせて住所情報を入力し、入力により近く一致する結果を出力として返します。</p>

## パラメータ

## 説明

## Option.HomeCountry

デフォルトの国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がドイツにある場合は、ドイツを指定します。**GetCandidateAddressLoqate** は、[StateProvince]、[PostalCode]、および [Country] の各住所フィールドから国を特定できなかった場合、指定された国を使用して、住所の検証を試みます。有効な国名には次のものがあります。

Afghanistan, Albania, Algeria, American Samoa, 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, Equatorial 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, Suriname, 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

## OutputCountryFormat

**Country** 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。

- E** 英語の国名を使用します (デフォルト)。
- I** 国名の代わりにその国の 2 文字の ISO の略語を使用します。
- U** 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。

パラメータ	説明
OutputScript	出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。 <b>Input</b> 書き直しを実行せず、入力と同じスクリプトで出力します(デフォルト)。 <b>Native</b> 使用可能な場合は、選択した国のネイティブ スクリプトで出力します。 <b>Latn</b> 英語の値を使用します。
Option.MaximumResults	GetCandidateAddressesLoqate が返す候補住所の最大数。デフォルトは 10 です。最大数は 99 です。

## 応答

GetCandidateAddressesLoqate は、次の出力を返します。

**表 24 : GetCandidateAddressesLoqate の出力**

応答要素	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。
AddressLine4	4 行目の住所行。
City	都市名。

応答要素	説明
Country	3 文字の ISO 3166-1 Alpha 3 国コード。ISO コードの一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。
FirmName	企業名。
PostalCode	郵便番号。米国では、ZIP Code™になります。
PostalCode.AddOn	ZIP + 4® Code の末尾 4 桁。米国住所にのみ適用されます。
ProcessedBy	住所を処理した住所コーダーを示します。 <b>LOQATE</b> Loqate コーダーが住所を処理しました。
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。次のいずれかの値になります。 • RequestFailed
Status.Description	問題の説明 (ある場合)。次のいずれかの値になります。 <b>Did not return multiples</b> 入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddressesLoqate は、一致する可能性のある住所が複数見つかった場合のみデータを返します。

## GetCityStateProvince

GetCityStateProvince は、与えられた入力郵便番号に対する都市および州/省を返します。



注：GetCityStateProvince は、米国およびカナダの住所のみをサポートします。

GetCityStateProvince は、Universal Addressing モジュールに含まれています。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

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

### XML 応答の例

次の例では、XML 応答を要求します。

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

この要求から次のような XML が返されます。

```
<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>

```

### 要求

#### 入力データのパラメータ

入力フィールドを以下の表に示します。

**表 25 : GetCityStateProvince の入力**

パラメータ	説明
Data.PostalCode	ZIP Code™ またはカナダの郵便番号を次のいずれかのフォーマットで示します。 99999 99999-9999 A9A9A9 A9A 9A9

## オプションのパラメータ

表 26 : GetCityStateProvince のオプション

パラメータ名	説明
Option.PerformUSProcessing	<p>米国住所を処理するかどうかを指定します。米国住所処理を有効にした場合、GetCityStateProvince は米国住所の州を返します。米国住所処理を無効にすると、米国住所は失敗します。つまり、Status 出力フィールドに "F" が設定されて返されます。出力フィールド Status.Code は、"DisabledCoder" となります。米国住所処理のライセンスを取得していない場合は、ジョブに米国住所が含まれるか否かにかかわらず、米国住所処理を無効にしなければ、ジョブを正常に実行することはできません。</p> <p>注：米国住所を正常に処理するには、米国住所処理の有効なライセンスを取得する必要があります。米国住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、米国住所処理を有効にすると、エラーとなります。</p> <p><b>Y</b>      はい、米国住所を処理します (デフォルト)。</p> <p><b>N</b>      いいえ、米国住所を処理しません。</p>
Option.Database.US	<p>米国住所処理に使用するデータベースを指定します。Management Console の [米国データベース リソース] パネルで定義されたデータベースのみが使用可能です。</p>
Option.PerformCanadianProcessing	<p>カナダ住所を処理するかどうかを指定します。カナダ住所処理を有効にした場合、GetCityStateProvince はカナダ住所の州を返します。カナダ住所処理を無効にした場合、Status フィールドに "F" が設定されカナダ住所は失敗します。出力フィールド Status.Code は、"DisabledCoder" となります。カナダ住所処理のライセンスを取得していない場合は、ジョブにカナダ住所が含まれるか否かにかかわらず、カナダ住所処理を無効にしなければ、ジョブを正常に実行することはできません。</p> <p>注：カナダ住所を正常に処理するには、カナダ住所処理の有効なライセンスを取得する必要があります。カナダ住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、カナダ住所処理を有効にすると、エラーとなります。</p> <p><b>Y</b>      カナダ住所を処理します (デフォルト)。</p> <p><b>N</b>      カナダ住所を処理しません。</p>

パラメータ名	説明
Option.Database.Canada	カナダ住所処理に使用するデータベースを指定します。Management Console の【カナダ データベース リソース】パネルで定義されたデータベースのみが使用可能です。
Option.OutputVanityCity	非正式な都市名を出力に含めるかどうかを指定します。非正式な都市名は、主要都市名の代替名です。例えば、Hollywood は Los Angeles の非正式な都市名です。 <b>Y</b> 非正式な都市名を含めます。 <b>N</b> 非正式な都市名を含めません (デフォルト)。
Option.MaximumResults	返される都市と州/省のペアの最大数を指定します。デフォルト値は 10 です。

## 応答

GetCityStateProvince は、入力郵便番号に一致する都市と州/省、およびマッチングの成功または失敗を示すコードを返します。複数の都市/州または都市/省が入力郵便番号にマッチする場合、複数の出力レコードが返されます。

表 27 : GetCityStateProvince の出力

応答要素	説明
City	一致した都市名。
City.Type	USPS® によって正規化された都市名のタイプ (米国住所のみ)。 <b>V</b> 非正式 (Non-Mailing) 都市名。 <b>P</b> 主要都市名。都市名は、主要な郵送都市名です。 <b>S</b> 補助的な都市名。都市名は、代替都市名ですが、許容されます。都市は複数の補助的な都市名を持つことができます。
PostalCode	入力郵便番号。

応答要素	説明
ProcessedBy	住所を処理した住所コーダーを示します。次のいずれかです。 <b>USA</b> 米国住所コーダーが住所を処理しました。 <b>CAN</b> カナダ住所コーダーが住所を処理しました。
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。次の値のみが有効です。 <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• UnrecognizedPostalCode</li> </ul>
Status.Description	失敗の説明。有効な値は次のとおりです。 <b>Postal code not found</b> Status.Code=UnrecognizedPostalCode の場合にこの値が表示されます。 <b>PerformUSProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。 <b>PerformCanadianProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。

### GetCityStateProvinceLoqate

GetCityStateProvinceLoqate は、与えられた入力郵便番号に対する都市および州/省を返します。このステージは、Universal Addressing モジュールに含まれています。

#### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

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

### XML 応答の例

次の例では、XML 応答を要求します。

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

この要求から次のような XML が返されます。

```
<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>
```

```
</ns2:output_port>
</ns2:xml.GetCityStateProvinceLoqateResponse>
```

## 要求

### 入力データのパラメータ

入力フィールドを以下の表に示します。

**表 28 : GetCityStateProvinceLoqate の入力**

パラメータ	説明
Data.Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2桁の ISO 国コード</li> <li>• 3桁の UPU 国コード</li> <li>• 英語の国名</li> </ul> <p>ISOコードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a> (735ページ) を参照してください。</p>
Data.PostalCode	住所の郵便番号。

## オプション

**表 29 : GetCityStateProvinceLoqate のオプション**

説明 / 有効な値
住所処理に使用するデータベースを指定します。Management Console の <b>[データベース リソース]</b> パネルで定義されたデータベースのみが使用可能です。
GetCityStateProvinceLoqate が返す必要がある住所の最大数。デフォルトは 10 です。

## 説明 / 有効な値

Option.OutputScript	出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。
<b>Input</b>	書き直しを実行せず、入力と同じスクリプトで出力します (デフォルト)。
<b>Native</b>	使用可能な場合は、選択した国のネイティブ スクリプトで出力します。
<b>Latn</b>	英語の値を使用します。

データ ライセンス エラーの発生時に Spectrum Technology Platform がどのように応答するかを指定します。

ジョブのエラー	データ ライセンス エラーが発生した場合、ジョブ全体をエラーにします。
レコードのエラー	データ ライセンス エラーの発生原因となったレコードをエラーにし、処理を続行します。

## 応答

GetCityStateProvinceLoqate は、入力郵便番号に一致する都市と州/省、およびマッチングの成功または失敗を示すコードを返します。複数の都市/州または都市/省が入力郵便番号にマッチする場合、複数の出力レコードが返されます。

表 30 : GetCityStateProvinceLoqate の出力

応答要素	説明
City	一致した都市名。
Country	で選択した、以下のいずれかのフォーマットで示された国。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>



応答要素	説明
PostalCode	入力郵便番号。
ProcessedBy	住所を処理した住所コーダーを示します。 <b>LOQATE</b> Loqate コーダーが住所を処理しました。
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。次の値のみが有効です。 • UnrecognizedPostalCode
Status.Description	失敗の説明。次の値のみが有効です。 <b>Postal code not found</b> Status.Code=UnrecognizedPostalCode の場合にこの値が表示されます。

## GetPostalCodes

GetPostalCodes では、特定の都市の郵便番号の検索が可能です。このサービスは、都市、州、および国を入力として受け取り、その都市の郵便番号を返します。入力を正しい順序で指定しなければ、郵便番号は返されません。

注：GetPostalCodes は、米国住所にのみ対応します。

GetPostalCodes は、Universal Addressing モジュールに含まれています。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

```
{ "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": []  
  }  
]}
```

### XML 応答の例

次の例では、XML 応答を要求します。

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

この要求から次のような XML が返されます。

```
<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>
```

### 要求

入力データのパラメータ

GetPostalCodes は、都市、州/省、および国を入力として受け取ります。

表 31 : GetPostalCodes の入力

パラメータ	説明
Data.City	郵便番号を検索する都市。 City フィールドに都市と州を入力できます。これを行う場合は、StateProvince フィールドを空白のままにする必要があります。 City および StateProvince フィールドの文字数が 100 文字を超えてはなりません。
Data.StateProvince	郵便番号を検索する都市の州または省。 州は、StateProvince フィールドではなく、City フィールドに入力することもできます。 City および StateProvince フィールドの文字数が 100 文字を超えてはなりません。
Data.Country	郵便番号を検索する都市の国コードまたは名前。有効な値は US のみです。

## オプションのパラメータ

表 32 : GetPostalCodes のオプション

パラメータ	説明
Option.Database.US	郵便番号検索に使用するデータベースを指定します。Management Console の [米国データベース リソース] パネルで定義されたデータベースのみが使用可能です。
Option.IncludeVanityCity	都市の非正式な都市名の郵便番号を含めるかどうかを指定します。非正式な都市名は、主要都市名の代替名です。例えば、Hollywood は Los Angeles の非正式な都市名です。 <b>Y</b> 非正式な都市名の郵便番号を含めます。 <b>N</b> 非正式な都市名の郵便番号を含めません (デフォルト)。

パラメータ	説明
Option.OutputCityType	都市タイプを出力で返すかどうかを指定します。有効にすると、都市タイプが City.Type フィールドに返されます。 <b>Y</b> 都市タイプを出力に含めます。 <b>N</b> 都市タイプを出力に含めません (デフォルト)。

## 応答

GetPostalCodes は、指定された都市の郵便番号を返します。各郵便番号は、以下の表に列挙されたデータとともにそれぞれ個別のレコードで返されます。

**表 33 : GetPostalCodes の出力**

応答要素	説明
City.Type	USPS® の都市タイプ (米国住所のみ)。都市タイプを判別するには、ZIP Code と都市名を調べます。例えば、メリーランド州ランハムの郵便番号は、20703、20706、および 20784 です。ランハムは、20703 と 20706 では主要都市ですが、20784 では非正式都市です。  このフィールド列に値が設定されるのは、Option.OutputCityType=Y の場合のみです。有効な値を次に示します。 <b>V</b> 非正式 (Non-Mailing) 都市名。 <b>P</b> 主要都市名。都市名は、主要な郵送都市名です。 <b>S</b> 補助的な都市名。都市名は、代替都市名ですが、許容されます。都市は複数の補助的な都市名を持つことができます。
PostalCode	指定された都市の郵便番号。
ProcessedBy	このサービスは米国住所に対してのみ機能するため、ProcessedBy には常に USA という 1 つの値が含まれます。

## 応答要素

## 説明

Status

マッチの成功または失敗。

<b>NULL</b>	成功
<b>F</b>	失敗

Status.Code

失敗の原因 (ある場合)。次のいずれかです。

- CountryNotSupported
- UnableToLookup

Status.Description

失敗の説明。

- 入力された国がサポートされていません
- 入力された都市が空白でした
- 入力された都市と州/省が空白であったか、一致が見つかりませんでした
- 都市と州の不一致 (スペルの相違が見つかるか、都市/州が非正式都市であるが、非正式マッチングが許可されていないか、都市/州が ZIP Code と一致しない)

`ValidateAddress` は、郵便当局の住所データを使用して、住所を正規化し、妥当性を確認します。`ValidateAddress` は、情報を修正し、管轄の郵便当局が推奨する書式で住所の書式を整えることができます。また、郵便番号、都市名、州/省名など、欠落している郵便情報を追加します。

`ValidateAddress` は、`ValidateAddress` が住所の妥当性を確認したかどうか、返した住所の確信レベル、住所の妥当性が確認できなかった場合はその理由など、バリデーション処理に関する結果インジケータも返します。

`ValidateAddress` は、住所のマッチングと正規化において、住所行をコンポーネントに分割し、それらを `Universal Addressing` モジュールの各種データベースの内容と比較します。マッチを検出した場合、入力住所をデータベース情報に合わせて正規化します。データベースにマッチしなかった場合、`ValidateAddress` は、オプションで入力住所の書式を整えます。書式設定プロセスでは、該当する郵便当局の規則に従って住所行の構成を試みます。

`ValidateAddress` は、`Universal Addressing` モジュールに含まれています。

## リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

```
{ "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": []
}] }
```

### XML 応答の例

次の例では、XML 応答を要求します。

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

この要求から次のような XML が返されます。

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

```

    <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>

```

### 要求

#### 入力データのパラメータ

`ValidateAddress` は、入力として住所を受け取ります。すべての住所は、その住所の国に関係無くこのフォーマットを使用します。米国住所に対する住所行データの処理方法に関する重要な情報については、「[米国住所の住所行処理 \(572ページ\)](#)」を参照してください。

表 34 : 入力フォーマット

パラメータ	書式	説明
<code>Data.AddressLine1</code>	文字列 [50]	最初の住所行。
<code>Data.AddressLine2</code>	文字列 [50]	2 行目の住所行。
<code>Data.AddressLine3</code>	文字列 [50]	3 行目の住所行。 カナダの住所には適用されません。
<code>Data.AddressLine5</code>	文字列 [50]	4 行目の住所行。 カナダの住所には適用されません。
<code>Data.AddressLine5</code>	文字列 [50]	5 行目の住所行。 英国住所にのみ適用されます。通り名、ユニット番号、ビルディング番号などを含めることができます。



パラメータ	書式	説明
Data.City	文字列 [50]	都市名。 米国住所に限り、都市、州、および ZIP Code™ を City フィールドに入力することができます。これを行う場合は、StateProvince フィールドと PostalCode フィールドを空白のままにする必要があります。
Data.StateProvince	文字列 [50]	州または省。 米国の住所に限り、州を StateProvince フィールドの代わりに City フィールドに入れることができます。
Data.PostalCode	文字列 [10]	住所に対する郵便番号を次のいずれかのフォーマットで示します。 99999 99999-9999 A9A9A9 A9A 9A9 9999 999  米国式の住所の場合に、住所に限り、ZIP Code™ を City フィールドに入力することができます。  米国式の住所の場合に、住所に限り、都市、州、または ZIP Code™ が PostalCode フィールドにある場合に、ValidateAddress がデータをパーシングして、住所を正しく処理することができます。最適な結果を得るため、このデータを適切なフィールド (City、StateProvince、PostalCode) に入力してください。
Data.Country	文字列 [50]	国コードまたは名前を、以下のいずれかのフォーマットで入力します。 <ul style="list-style-type: none"> <li>• 2 文字の ISO 3166-1 Alpha 2 国コード</li> <li>• 3 文字の ISO 3166-1 Alpha 3 国コード</li> <li>• 英語の国名</li> <li>• フランス語の国名</li> <li>• ドイツ語の国名</li> <li>• スペイン語の国名</li> </ul> ISO コードの一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735 ページ) を参照してください。
Data.FirmName	文字列 [50]	会社名または企業名。

パラメータ	書式	説明
Data.USUrbanName	文字列 [50]	米国住所都市化名。主にプエルトリコの住所で使用されます。
Data.CustomerID	文字列 [9]	この郵便物が汎用バーコードを使用している場合、USPS®が割り当てた顧客 ID をこのフィールドに指定します。ValidateAddress の汎用バーコードは、OneCode ACS®サービスを使用する郵便物に使用されます。
Data.CanLanguage	文字列	カナダの住所に限り、Option.CanFrenchFormat=T オプションが使用されている場合に、住所が英語かフランス語かを示します。  このフィールドが空白の場合、アドレスは英語でフォーマットされています。このフィールドに空白以外の値が含まれる場合、住所はフランス語でフォーマットされています。ケベックの住所は、このフィールドの値に関係なく常にフランス語でフォーマットされます。

## 米国住所の住所行処理

米国住所の場合、AddressLine1 から AddressLine4 の入力フィールドの処理方法は、企業名抽出または都市化コード抽出のオプションが有効になっているかどうかによって異なります。2つのオプションのいずれかが有効になっている場合は、ValidateAddress は、4つすべてのフィールドのデータを参照して、住所の妥当性を確認し、要求されたデータ (企業名または都市化コード) を抽出します。どちらのオプションも有効でない場合は、ValidateAddress は、空白でない最初の2つの住所行フィールドのみを使用して、妥当性を確認します。他の住所行フィールドのデータは、AdditionalInputData 出力フィールドに返されます。例を次に示します。

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

この住所において、企業名抽出または都市化コード抽出が有効である場合は、ValidateAddress は、4つすべての住所行を確認します。企業名抽出と都市化コード抽出がどちらも有効でない場合は、ValidateAddress は、AddressLine1 と AddressLine3 (空白でない最初の2つの住所行) を参照して、そのデータを使用して住所の妥当性を確認します。AddressLine4 のデータは、AdditionalInputData 出力フィールドに返されます。

オプションのパラメータ  
出力データ オプション

以下の表に、ValidateAddress が返す情報の種類を制御するオプションの一覧を示します。これらのオプションのうちの一部は、カナダ住所に対してオーバーライドすることができます。詳細については、[カナダ住所のオプション](#)（604ページ）を参照してください。

表 35 : 出力データ オプション

パラメータ	説明
Option.OutputRecordType	<p>出力レコードのタイプ。1つ以上の場合はリストで提供されます。</p> <ul style="list-style-type: none"> <li data-bbox="690 661 1421 1060"><b>A</b> 住所データの1～4行に加えて、都市、州、郵便番号、企業名、および都市化名情報を返します。各住所行は、封筒に記載される住所の実際の行に対応しています。詳細については、<a href="#">応答</a>（616ページ）を参照してください。住所の妥当性を確認できた場合は、住所行には正規化済み住所が含まれます。正規化済み住所では、句読文字が取り除かれ、方向指示と通り接尾語には省略形が使用され、住所要素が修正されています。住所の妥当性を確認できなかった場合は、住所行には入力住所がそのまま含まれます ("パス スルー" データ)。OutputRecordType=A を指定しなかった場合でも、妥当性が確認されなかった住所は必ず、パス スルー データとして住所行フィールドに含まれます。</li> <li data-bbox="690 1071 1421 1281"><b>E</b> パース済み住所要素。家番号、通り名、通り接尾語、方向指示などの住所の各要素が、個別のフィールドに返されます。詳細については、<a href="#">#unique_113</a>を参照してください。"E" を指定し、OutputFormattedOnFail=Y を指定した場合は、パース済み住所要素に、妥当性が確認できなかった住所の入力住所が含まれることになります。</li> <li data-bbox="690 1291 1421 1659"><b>I</b> パース済み入力。このオプションでは、住所の妥当性を確認できたかどうかにかかわらず、入力住所をパース済み形式で返します。家番号、通り名、通り接尾語、方向指示などの入力住所の各要素が、個別のフィールドに返されます。パース済み入力 (値 "I") は、妥当性を確認できなかった入力だけでなく、すべての入力住所をパース済み形式で返す点で、OutputRecordType=E と OutputFormattedOnFail=Y の組み合わせとは異なります。詳細については、<a href="#">パース済み入力</a>（620ページ）を参照してください。</li> <li data-bbox="690 1669 1421 1774"><b>P</b> 郵便データ。出力住所には、妥当性が確認された各住所の追加データが含まれます。詳細については、<a href="#">郵便データ出力</a>（623ページ）を参照してください。</li> </ul> <p>空 住所データまたは郵便データを返しません。 白</p>

## パラメータ

## 説明

---

**Option.OutputFieldLevelReturnCodes**

フィールドレベルの結果インジケータを含めるかどうかを指定します。フィールドレベルの結果インジケータは、各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、HouseNumber のフィールドレベルの結果インジケータは **HouseNumber.Result** に格納されます。結果インジケータの出力フィールドの完全な一覧は、[フィールドレベルの結果インジケータ \(629ページ\)](#) を参照してください。

- N** フィールドレベルのリターン コードを出力しません (デフォルト)。
  - Y** フィールドレベルのリターン コードを出力します。
-

パラメータ

説明

---

Option.OutputFormattedOnFail

## パラメータ

## 説明

住所の妥当性を確認できない場合に書式を整えた住所を返すかどうかを指定します。住所には、その国の標準住所書式が設定されます。このオプションを選択しない場合、住所の妥当性を確認できないと、出力住所フィールドは空白になります。

注：このオプションは、米国およびカナダの住所のみに適用されます。その他の住所に対して書式を整えたデータは返されません。

**N** 失敗した住所の書式を整えません (デフォルト)。

**Y** 失敗した住所の書式を整えます。

OutputRecordType オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。OutputRecordType=E を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。ValidateAddress が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、OutputRecordType=I を指定します。

Option.OutputRecordType オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。Option.OutputRecordType=E を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。ValidateAddress が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、Option.OutputRecordType=I を指定します。

**[標準住所を含める]**、**[住所行の要素を含める]**、および **[郵便情報を含める]** の各チェック ボックスで指定されたフォーマットを使用して、書式を整えた住所が返されます。**[住所行の要素を含める]** を選択した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。ValidateAddress が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**[正規化された入力住所要素を含める]** を選択します。

Y を指定する場合は、OutputRecordType に対して "A" と "E" またはそのいずれかを指定する必要があります。

## パラメータ

## 説明

Y を指定する場合は、Option.OutputRecordType に対して "A" と "E" またはそのいずれかを指定する必要があります。

このオプションをオンにする場合は、**[標準住所を含める]** と **[住所行の要素を含める]** またはそのいずれかを選択する必要があります。

## Option.OutputStreetNameAlias

米国住所に対しては、通り名のエイリアスを出力に使用するかどうかを指定します。通り名のエイリアスとは、通りに対する別名で、通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。基本名とは、通り全体に適用される名称です。

- N** 出力において、通り名のエイリアスを返しません。
- Y** 通り名のエイリアスが存在する場合は、出力において、通り名のエイリアスを返します (デフォルト)。

## Option.OutputStreetNameAlias

米国住所に対しては、入力に使用されている通り名のエイリアスの処理方法を指定します。通り名のエイリアスとは、通りに対する別名で、通常は、通り上の特定の範囲の住所のみに対して使用されます。

このオプションを有効にすると、入力に使用されている通り名のエイリアスが、出力にも使用されます。このオプションを有効にしない場合は、入力に使用されている通り名のエイリアスが、出力では基本通り名に変換されますが、次の例外があります。

- 入力において、よく使用されるエイリアスが使用されている場合は、そのエイリアスが必ず出力で使用されます。
- 入力で使用されている変更名のエイリアスは、出力では必ず基本通り名に変換されます。

これは、ValidateAddress で通り名のエイリアスを処理する方法を制御する 3 つのオプションのうちの 1 つです。他の 2 つは Option.OutputPreferredAlias と Option.OutputAbbreviatedAlias です。

注: Option.OutputAbbreviatedAlias を有効にした場合は、Option.OutputStreetNameAlias を無効にした場合でも、必ず省略形エイリアスが出力に使用されます。

- N** 出力において、通り名のエイリアスを返しません。
- Y** 入力された通り名がエイリアスである場合は、出力において、通り名のエイリアスを返します (デフォルト)。

## パラメータ

## 説明

---

**Option.OutputAddressBlocks**

実際の郵便物に印字される、書式を整えた住所を返すかどうかを指定します。住所の各行が、別々の住所ブロックフィールドに入れて返されます。**AddressBlock1** から **AddressBlock9** まで、最大 9 つの住所ブロック出力フィールドが使用されます。

例えば、以下の住所入力の場合、

AddressLine1: 4200 Parliament Place  
AddressLine2: Suite 600  
City: Lanham  
StateProvince: MD  
PostalCode: 20706

以下の住所ブロックが出力されます。

AddressBlock1: 4200 PARLIAMENT PL STE 600  
AddressBlock2: LANHAM MD 20706-1882  
AddressBlock3: UNITED STATES OF AMERICA

**ValidateAddress** は、郵便当局の規格に従って、住所の書式を整えて、住所ブロックの形式にします。国名は、万国郵便連合 (UPU) の国名で返されます。**Option.OutputCountryFormat** オプションは、住所ブロックの国名には影響を与えないことに注意してください。このオプションは、**Country** 出力フィールドに返される名前のみに影響を与えます。

米国およびカナダ以外の住所に対しては、**ValidateAddress** で住所の妥当性が確認できなかった場合、住所ブロックは返されません。米国およびカナダの住所に対しては、妥当性が確認できなかった場合も住所ブロックが返されます。

- N** 住所ブロックを返しません。こちらがデフォルトです。
  - Y** 住所ブロックを返します。
-



## パラメータ

## 説明

## Option.OutputAMAS

実際の郵便物に印字される、書式を整えた住所を返すかどうかを指定します。住所の各行が、別々の住所ブロックフィールドに入れて返されます。**AddressBlock1** から **AddressBlock9** まで、最大 9 つの住所ブロック出力フィールドが使用されます。

例えば、以下の住所入力の場合、

AddressLine1: 4200 Parliament Place  
AddressLine2: Suite 600  
City: Lanham  
StateProvince: MD  
PostalCode: 20706

以下の住所ブロックが出力されます。

AddressBlock1: 4200 PARLIAMENT PL STE 600  
AddressBlock2: LANHAM MD 20706-1882  
AddressBlock3: UNITED STATES OF AMERICA

**ValidateAddress** は、郵便当局の規格に従って、住所の書式を整えて、住所ブロックの形式にします。国名は、万国郵便連合 (UPU) の国名で返されます。**Option.OutputCountryFormat** オプションは、住所ブロックの国名には影響を与えないことに注意してください。このオプションは、**Country** 出力フィールドに返される名前のみに影響を与えます。

米国およびカナダ以外の住所に対しては、**ValidateAddress** で住所の妥当性が確認できなかった場合、住所ブロックは返されません。米国およびカナダの住所に対しては、妥当性が確認できなかった場合も住所ブロックが返されます。

- N** 住所ブロックを返しません。こちらがデフォルトです。
- Y** 住所ブロックを返します。

## 下院選挙区の取得

**ValidateAddress** は、住所に対する米下院選挙区を特定できます。

下院選挙区を取得するには、**Option.OutputRecordType** に **P** を含める必要があります。**Option.OutputRecordType** の詳細については、[出力データオプション \(573ページ\)](#) を参照してください。

表 36 : 下院選挙区出力

応答要素	説明
USCongressionalDistrict	下院選挙区番号。住所が州以外の住所 (プエルトリコやワシントン D.C. など) である場合は、このフィールドは空白になります。

## 郡名の取得

ValidateAddress は、ある住所が所在する郡を特定し、その郡名を返すことができます。

注：郡名は、米国住所にのみ適用されます。

郡名を取得するには、Option.OutputRecordType に P を含める必要があります。Option.OutputRecordType の詳細については、[出力データオプション \(573ページ\)](#) を参照してください。

表 37 : 郡名出力

応答要素	説明
USCountyName	郡名

## FIPS 郡番号の取得

連邦情報処理標準 (FIPS) 郡番号は、州の中の各郡を識別する番号です。これらの番号は、州レベルにおいてのみ一意であり、国レベルでは一意ではないことに注意してください。詳細については、<http://www.census.gov> を参照してください。

注：FIPS 郡番号は、米国住所にのみ適用されます。

FIPS 郡番号を取得するには、Option.OutputRecordType に P を含める必要があります。Option.OutputRecordType の詳細については、[出力データオプション \(573ページ\)](#) を参照してください。

表 38 : FIPS 郡番号出力

応答要素	説明
USFIPSCountyNumber	FIPS (連邦情報処理標準) 郡番号

#### 配達ルート コードの取得

配達ルート コードとは、個々の郵便配達者に割り当てられた一意の識別子で、これによって米国の各配達ルートを一意に識別することができます。ValidateAddress は、宛先の配達ルートを表すコードを返すことができます。

注：配達ルート コードは、米国住所に対してのみ存在します。

配達ルート コードを取得するには、Option.OutputRecordType に P を含める必要があります。Option.OutputRecordType の詳細については、[出力データオプション \(573ページ\)](#) を参照してください。

表 39 : 配達ルート コード出力

応答要素	説明
USCarrierRouteCode	配達ルート コード

#### 配達ポイント バーコードの作成

配達ポイント バーコード (DPBC) は、住所を POSTNET™ バーコードで表記したものです。開始および終了フレームバーと、ZIP+4® Code、ストリートの住所の番号に基づいて計算された値、および修正ディジットの 1 桁ごとに 5 本のバーがあり、合計 62 本のバーで構成されます。DPBC により、配達業者の徒歩経路のレベルにまで手紙を自動仕分けすることができます。ValidateAddress は、DPBC の作成に必要なデータを生成します。

注：配達ポイント バーコードは、米国住所に対してのみ存在します。配達ポイント バーコードの詳細については、<http://www.usps.com> を参照してください。

DPBC の作成に必要なデータを生成するには、Option.OutputRecordType に P を含める必要があります。Option.OutputRecordType の詳細については、[出力データオプション \(573ページ\)](#) を参照してください。

表 40 : 配達ポイント バーコード出力

応答要素	説明
PostalBarCode	配達ポイント バーコードの配達ポイント部分
USBCCheckDigit	11桁の配達ポイント バーコードのチェック デジット部分

DPBC を作成するには、ValidateAddress 出力の値を次のように結合します。

PostalCode.Base + PostalCode.Addon + PostalBarcode + USBCCheckDigit

例えば、次のデータがあるとしてします。

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

このデータから作成されるバーコードは、次のようになります。

494234506292

#### デフォルト オプション

以下の表に、住所のフォーマットと処理を制御するオプションの一覧を示します。これらのオプションはデフォルトですべての住所に適用されるので、"デフォルト オプション" と言います。これらのオプションのうちの一部は、カナダ住所に対してオーバーライドすることができます。詳細については、「[カナダ住所のオプション](#) (604ページ)」を参照してください。

表 41 : デフォルト オプション

パラメータ	説明
Option.OutputCasing	出力住所の大文字と小文字の区別を指定します。次のいずれかです。 <ul style="list-style-type: none"> <li><b>M</b> 出力に大文字と小文字を混在させます (デフォルト)。例: 123 Main StMytown FL 12345</li> <li><b>U</b> 出力に大文字を使用します。例: 123 MAIN STMYTOWN FL 12345</li> </ul>

パラメータ	説明
Option.OutputPostalCodeSeparator	<p>ZIP™ Code またはカナダの郵便番号に区切り文字 (スペースまたはハイフン) を使用するかどうかを指定します。</p> <p>例えば、区切り文字ありの ZIP + 4® Code は 20706-1844、区切り文字なしは 207061844 になります。区切り文字ありのカナダの郵便番号は P5E"1S7、区切り文字なしは P5E1S7 になります。</p> <p><b>Y</b> 区切り文字を使用します (デフォルト)。</p> <p><b>N</b> 区切り文字を使用しません。</p> <p>注：カナダの郵便番号ではスペースが、米国の ZIP + 4® コードではハイフンが使用されます。</p>
Option.OutputMultinationalCharacters	<p>ウムラウト記号やアクセント記号などの付加記号を含む多国籍文字を返すかどうかを指定します(米国住所ではサポートされません)。</p> <p><b>N</b> 出力に多国籍文字を使用しません(デフォルト)。標準の ASCII 文字のみが返されます。</p> <p><b>Y</b> 出力に多国籍文字を使用します。</p>
Option.KeepMultimatch	<p>一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを示します。</p> <p><b>Y</b> 複数のマッチを返します (デフォルト)。</p> <p><b>N</b> 複数のマッチを返しません。</p> <p>詳細については、「<a href="#">複数マッチを返す (589ページ)</a>」を参照してください。</p>

## パラメータ

## 説明

## Option.StandardAddressFormat

米国住所の補助的な住所情報を配置する場所を指定します。補助的な住所情報とは、部屋番号やアパート番号などの指定子のことです。例えば、次の住所の補助的な住所情報は "Apt 10E" で、主要な住所情報は "424 Washington Blvd" です。

Apt 10E424 Washington BlvdSpringfield MI 49423

- C** 主要な住所情報と補助的な住所情報の両方を AddressLine1に配置します (デフォルト)。
- S** 主要な住所情報を AddressLine1に、補助的な住所情報を AddressLine2 に配置します。
- D** 主要な住所情報と補助的な住所情報の両方を AddressLine1に配置し、二重住所からドロップされた情報を AddressLine2 に配置します。二重住所とは、通り情報と、PO Box/地方配送路/幹線請負契約情報の両方を含む住所のことです。詳細については、「[二重住所ロジックについて \(588ページ\)](#)」を参照してください。

## Option.OutputShortCityName

短い都市名または非正式な都市名を代替名として持つ都市名を書式設定する方法を指定します。米国およびカナダの住所に適用されます。

- Y** USPS® が承認した都市の略称が 1 つある場合、それを返します。USPS® は、14 文字以上の都市名に対し、略称を定めています。都市の略称は 13 文字以下で、宛名ラベルのサイズが限られている場合に使用できます。短い都市名が存在しない都市に対しては、正式な都市名が返されます。
- N** 長い都市名を返します (デフォルト)。
- S** 入力住所に省略された都市名が使用されている場合にのみ、省略された都市名を返します。入力住所に短い都市名が使用されていない場合は、その都市に対する USPS® の規則によって、長い都市名または短い都市名が返されます。CASS™ テストを実行する場合は、このオプションを選択します。
- V** 入力都市名が非正式な都市名の場合、非正式な都市名 (非正式名) を出力します。例えば、"Hollywood" は "Los Angeles" の非正式な都市名です。このオプションを選択せず、入力都市名が非正式な都市名の場合、長いバージョンの郵送都市名が返されます。

## パラメータ

## 説明

---

**Option.OutputCountryFormat**

**Country** 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。

- E** 英語の国名を使用します (デフォルト)。
  - S** スペイン語の国名を使用します。
  - F** フランス語の国名を使用します。
  - G** ドイツ語の国名を使用します。
  - I** 国名の代わりにその国の2文字のISOの略語を使用します。
  - U** 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。
-

## パラメータ

## 説明

## Option.HomeCountry

デフォルトの国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がカナダにある場合は、カナダを指定します。ValidateAddress は、[StateProvince]、[PostalCode]、および [Country] の各住所フィールドから国を特定できなかった場合、指定された国を使用して、住所の検証を試みます。有効な国名には次のものがあります。

Afghanistan, Albania, Algeria, American Samoa, 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, Equatorial 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, Suriname, 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



## パラメータ

## 説明

## Option.DualAddressLogic

複数の空白ではない住所行があるか、複数の住所タイプが同じ住所行に設定されている場合にどのタイプの一致を返すかを指定します。(米国住所のみ)。

**N** (デフォルト)USPS®CASS™ の規則では、以下の優先順位に基づいて、返す住所を決定します。

1. PO Box
2. Firm
3. Highrise
4. Street
5. Rural Route
6. General Delivery

**S** 住所行に関係なく、通り一致を返します。

**P** 住所行に関係なく、PO Box 一致を返します。

詳細については、「[二重住所ロジックについて \(588ページ\)](#)」を参照してください。

## 二重住所ロジックについて

米国住所に対してのみ、Option.DualAddressLogic オプションは、通り情報と PO Box/地方配送路/幹線請負契約情報の両方が住所に含まれる場合に、Validate Address が通り一致を返すか、または PO Box/地方配送路/幹線請負契約一致を返すかを制御します。

注：通り情報が PO Box/地方配送路/幹線請負契約情報と別の住所行入力フィールドに含まれている場合、Option.DualAddressLogic オプションは、効果を持ちません。

例えば、次の入力住所が与えられたとします。

AddressLine1: 401 N Main St Apt 1 POB 1City: KempStateProvince: TXPostalCode: 75143

ValidateAddress は、次のいずれかを返します。

- Option.DualAddressLogic が N または P のいずれかに設定されている場合は、次を返します。

AddressLine1: PO Box 1City: KempStateProvince: TXPostalCode: 75143-0001

- Option.DualAddressLogic が S に設定されている場合は、次を返します。

AddressLine1: 401 N Main St Apt 1City: KempStateProvince: TXPostalCode: 75143-4806

住所の正規化に使用されない住所データは、次の 2 カ所のいずれかに返すことができます。

- **AddressLine2 — Option.StandardAddressFormat=D** を指定した場合、住所の正規化に使用されない住所情報は AddressLine2 フィールドに返されます。詳細については、[デフォルト オプション](#) (583ページ) を参照してください。例えば、二重住所に対して通り一致を返すと選択すると、次のようになります。

AddressLine1: 401 N Main St Apt 1AddressLine2: PO Box 1City: KempStateProvince:  
TXPostalCode: 75143-0001

- **AdditionalInputData** — Option.StandardAddressFormat=D を指定しない場合、住所の正規化に使用されない住所情報は **AdditionalInputData** フィールドに返されます。このオプションの詳細については、[デフォルト オプション](#) (583ページ) を参照してください。例えば、二重住所に対して通り一致を返すと選択すると、次のようになります。

AddressLine1: 401 N Main St Apt 1City: KempStateProvince: TXPostalCode:  
75143-0001AdditionalInputData: PO Box 1

ドロップされた住所情報を取得するには、Option.StandardAddressFormat オプションを D に設定します。詳細については、[デフォルト オプション](#) (583ページ) を参照してください。

複数マッチを返す

ValidateAddress が、入力住所に一致する可能性のある複数の住所を郵便データベース内で検出した場合、一致する可能性のある住所を ValidateAddress が返すよう設定できます。例えば、次の住所は米国郵便データベース内の複数の住所にマッチします。

PO BOX 1  
New York, NY

## オプション

複数マッチを返すには、次の表に示すオプションを使用します。

表 42: 複数マッチのオプション

パラメータ	説明
Option.KeepMultimatch	一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを示します。
<b>Y</b>	複数のマッチを返します (デフォルト)。
<b>N</b>	複数のマッチを返しません。

パラメータ	説明
Option.MaximumResults	<p>返す住所の最大数を示す 1 ~ 10 の数字を入力します。</p> <p>デフォルト値は 1 です。</p> <p>注： Option.Keepmultimatch=N と Option.KeepMultimatch=Y/Option.MaximumResults=1 の違いは、 Option.KeepMultimatch=N は複数マッチによって失敗が返され、 Option.KeepMultimatch=Y かつ Option.MaximumResults=1 は複数マッチによって 1 つのレコードが返される点です。</p>
Option.OutputFieldLevelReturnCodes	<p>どの出力住所が候補住所かを特定するには、 Option.OutputFieldLevelReturnCodes に対して値 <b>Y</b> を指定する必要があります。このように設定すると、候補住所のレコードのフィールドレベルの結果インジケータに 1 つ以上の値 "M" が格納されます。</p>

## 出力

複数マッチを返すよう選択した場合、住所は指定した住所フォーマットで返されます。住所フォーマットの指定については、[出力データオプション \(573ページ\)](#) を参照してください。どのレコードが候補住所であるかを特定するには、フィールドレベルの結果インジケータに複数の値 "M" があるかどうかを調べます。詳細については、[フィールドレベルの結果インジケータ \(629ページ\)](#) を参照してください。

## 米国住所のオプション

パラメータ	説明
Option.PerformUSProcessing	<p>米国住所を処理するかどうかを指定します。米国住所処理を有効にすると、<b>ValidateAddress</b> は米国住所の検証を試みます。米国住所処理を無効にすると、米国住所は失敗します。つまり、<b>Status</b> 出力フィールドに "F" が設定されて返されます。出力フィールド <b>Status.Code</b> は、"DisabledCoder" となります。米国住所処理のライセンスを取得していない場合は、ジョブに米国住所が含まれるか否かにかかわらず、米国住所処理を無効にしなければ、ジョブを正常に実行することはできません。</p> <p>注：米国住所を正常に処理するには、米国住所処理の有効なライセンスを取得する必要があります。米国住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、米国住所処理を有効にすると、エラーとなります。</p> <p><b>N</b>      いいえ、米国住所を処理しません。</p> <p><b>Y</b>      はい、米国住所を処理します。デフォルト</p>
Option.Database.US	<p>米国住所の妥当性の確認に使用するデータベースを指定します。<b>Management Console</b> の [米国データベース リソース] パネルで定義されたデータベースのみが使用可能です。</p>
Option.PerformLOT	<p>Enhanced Line of Travel (eLOT) 処理は、トラベルライン シーケンス コードを住所に割り当てます。住所は eLOT 順でソートされませんが、住所を eLOT 順にソートするために使えるトラベルライン シーケンス コードが提供されます。</p> <p>eLOT 処理を実行するには、eLOT データベースをインストールしておく必要があります。</p> <p><b>N</b>      トラベルライン処理を実行しません。デフォルト</p> <p><b>Y</b>      トラベルライン処理を実行します。</p> <p>このオプションで返される出力フィールドの一覧については、<a href="#">Enhanced Line of Travel 出力 (647ページ)</a> を参照してください。</p>

パラメータ	説明
Option.PerformRDI	<p>Residential Delivery Indicator (RDI™) 処理は、住所が (企業住所ではなく) 個人住所であることを調べます。RDI™ 処理を実行するには、RDI™ データベースをインストールしておく必要があります。</p> <p>DPV® と RDI™ の両方の処理を有効にしている場合は、RDI™ 情報は、住所が有効な配達ポイントである場合のみ返されます。DPV® で住所の妥当性が確認されなかった場合は、RDI™ データは返されません。</p> <p><b>N</b> Residential Delivery Indicator 処理を実行しません。デフォルト</p> <p><b>Y</b> Residential Delivery Indicator 処理を実行します。</p>
Option.PerformESM	<p>Enhanced Street Matching (ESM) は、追加のマッチング ロジックを適用することにより、綴りに誤りがある通り名や複雑な通り名を修正し、マッチ結果を得ます。ESMを使うと、より多くの住所の妥当性を確認できるようになりますが、パフォーマンスは低下します。ASMが有効な場合は、ESMを実行することはできません。</p> <p><b>N</b> Enhanced Street Matching を実行しません。デフォルト</p> <p><b>Y</b> Enhanced Street Matching を実行します。</p>
Option.PerformASM	<p>All Street Matching (ASM) は、ESM 処理に加えて追加のマッチング ロジックを適用することにより、通り名の誤りを修正し、マッチ結果を得ます。通りの最初の文字が誤っている場合の通りのマッチングに有効です。ASMは、最良の住所検証結果を提供しますが、パフォーマンスは低下します。</p> <p><b>N</b> All Street Matching を実行しません。</p> <p><b>Y</b> All Street Matching を実行します。デフォルト</p>

## パラメータ

## 説明

## Option.PerformDPV

Delivery Point Validation (DPV<sup>®</sup>) は、特定の住所が有効な住所の範囲内にあるかどうかを確認するのではなく、特定の住所が存在するかどうかを確認します。CMRA 処理は、住所が、Commercial Mail Receiving Agency (CMRA: 民間私書箱) と呼ばれる民間企業が貸し出す私書箱であるかどうかを確認します。

DPV および CMRA 処理を実行するには、DPV データベースをインストールしておく必要があります。DPV データベースには、DPV と CMRA の両方のデータが含まれます。

**N** Delivery Point Validation または CMRA 処理を実行しません。デフォルト

**Y** Delivery Point Validation または CMRA 処理を実行します。

このオプションで返される出力フィールドの一覧については、[DPV および CMRA 出力](#) (650ページ) を参照してください。

## Option.PerformLACSLink

USPS<sup>®</sup> Locatable Address Conversion System (LACS) は、地方配送路の住所を通り名に沿った住所に変換した場合、PO Box 番号の再割り当てがあった場合、または通り名に沿った住所が変更した場合に、それに伴って変更した住所を修正します。LACS<sup>Link</sup> 処理を有効にした場合、妥当性が確認できなかった住所、または妥当性が確認され、LACS<sup>Link</sup> 変換のフラグが付けられた住所に対し、その処理が実行されます。

LACS<sup>Link</sup> 処理を実行するには、LACS<sup>Link</sup> データベースをインストールしておく必要があります。

**N** LACS<sup>Link</sup> 変換を行いません。デフォルト

**Y** LACS<sup>Link</sup> 変換を行います。

このオプションで返される出力フィールドの一覧については、[LACSLink 出力](#) (648ページ) を参照してください。

## パラメータ

## 説明

---

**Option.PerformEWS**

Early Warning System (EWS) は、USPS® EWS ファイルを使用して、ZIP + 4® データベースには存在しない住所の妥当性を確認します。

EWS 処理を実行するには、EWS データベースをインストールしておく必要があります。

入力住所が、EWS ファイルの住所に一致する場合、次のレコード レベルの結果インジケータが返されます。

- Status="F"
- Status.Code="EWSFailure"
- Status.Description="Address found in EWS table"

**N** EWS 処理を実行しません。デフォルト

**Y** EWS 処理を実行します。

---

パラメータ

説明

---

Option.ExtractFirm



## パラメータ

## 説明

AddressLine1 ~ AddressLine4 から企業名を抽出し、FirmName 出力フィールドに入れるかどうかを指定します。このオプションは、入力レコードの FirmName フィールドが空白で、住所行が複数存在する場合に適用されます。

- Y**            企業名を抽出します。
- N**            企業名を抽出しません。デフォルト

住所行の中の企業名を特定するため、住所行をスキャンし、どのフィールドが住所行で、どのフィールドが企業名の行であるかを特定するためのキーワードおよびパターンが検索されます。この処理はパターンに基づいて行われるため、フィールドが誤って認識される場合があります。最適な企業名抽出を行うためのヒントを、以下に示します。

- 可能ならば、主要な住所要素を AddressLine1、補助的な要素を AddressLine2、都市化名を AddressLine3、企業名を AddressLine4 に配置します。住所に都市化コードが存在しない場合は、企業名を AddressLine3 に配置し、AddressLine4 を空白にします。例を次に示します。

**AddressLine1:** 4200 Parliament Place**AddressLine2:** Suite 600**AddressLine3:** Pitney Bowes  
**AddressLine4:** <blank>

- 住所行を 2 行だけ定義する場合は、ほとんどの場合 AddressLine2 には補助的な住所が入ります。AddressLine2 を企業名として処理する確率を上げるには、企業名を AddressLine3 に配置し、AddressLine2 は空白にします。
- 企業名に数字が含まれていると ("1 Stop Software" の "1" など)、そのフィールドが住所行として扱われる可能性が高くなります。

以下に、企業名抽出の例をいくつか示します。

- 次の例では、AddressLine2 が FirmName 出力フィールドに抽出されます。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place Suite 600**AddressLine2:** International Goose Feathers inc.

- 次の例では、AddressLine3 が FirmName 出力フィールドに抽出されます。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place**AddressLine2:** Suite 600**AddressLine3:** Pitney Bowes

- 次の例では、AddressLine3 は AdditionalInputData 出力フィールドに配置されます。FirmName 入力フィールドが空白ではないため、企業名は抽出されません。

**FirmName:** International Goose Feathers Inc.**AddressLine1:** 4200 Parliament Place**AddressLine2:** Suite 600**AddressLine3:** Pitney Bowes

- 次の例では、空白でない住所行が 1 行しかなく、その行は必ず主要な住所要素として処理されるため、企業名は抽出されません。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place Suite 600

## パラメータ

## 説明

- 次の例では、AddressLine2は補助的な住所要素として処理されます。数字の"1"が含まれているために、そのフィールドが補助的な住所要素として扱われるためです。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place Suite  
600**AddressLine2:** 1 Stop Software

## Option.ExtractUrb

AddressLine1～AddressLine4から都市化名を抽出し、USUrbanName出力フィールドに入れるかどうかを指定します。このオプションは、入力レコードのUSUrbanNameフィールドが空白で、住所行が複数存在する場合に適用されます。

**Y** 都市化名を抽出します。

**N** 都市化名を抽出しません。デフォルト

住所行の中の都市化名を特定するため、住所行をスキャンし、どのフィールドが住所行で、どのフィールドが都市化名の行であるかを特定するためのキーワードおよびパターンが検索されます。この処理はパターンに基づいて行われるため、フィールドを誤って認識する場合があります。最適な都市化名抽出を行うには、できる限り、主要な住所要素をAddressLine1、補助的な要素をAddressLine2、都市化名をAddressLine3、企業名をAddressLine4に配置します。例を次に示します。

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

## パラメータ

## 説明

## Option.PerformSuiteLink

Suite<sup>Link™</sup> 処理を実行するかどうかを指定します。

Suite<sup>Link</sup> は、補助的な住所情報の妥当性が確認できなかった米国の企業住所に対し、その補助的な住所情報を修正します。Suite<sup>Link</sup> 処理が有効になっている場合、既知の企業名とその補助的な住所情報からなるデータベースに対して、企業名のマッチングが行われます。

例を次に示します。

企業名: Pitney Bowes  
住所行 1: 4200 Parliament Place  
Address Line 2: STE 1  
郵便番号: 20706

この場合、Suite<sup>Link</sup> は、Suite 番号を以下の正しい Suite 番号に変更します。

企業名: Pitney Bowes  
住所行 1: 4200 Parliament Place  
Address Line 2: **STE 600**  
Postal Code: 20706-1844

Suite<sup>Link™</sup> 処理を実行するには、Suite<sup>Link™</sup> データベースをインストールしておく必要があります。

このオプションは、次の値のいずれかを取ります。

- N** Suite<sup>Link™</sup> を使用しません。デフォルト
- Y** Suite<sup>Link™</sup> を使用します。

このオプションで返されるフィールドの一覧については、[SuiteLink 出力 \(652ページ\)](#) を参照してください。

## パラメータ

## 説明

## Option.OutputPreferredAlias

通り名に対してよく使用されるエイリアスを出力に使用するかどうかを指定します。

米国における通り名のエイリアスとは、通りの一部に付けられた別名のことです。通り名のエイリアスには、次の 4 種類があります。

- **よく使用される名前** — その地域でよく使用される通り名です。通常は、通り上の特定の範囲の住所のみに対して使用されます。
- **省略形** — 通り名の省略形です。AddressLine1 の長さが 31 文字以上になる場合に使用することができます。例えば、1234 BERKSHIRE VALLEY RD APT 312A という通り名は、1234 BERKSHIRE VLLY RD APT 312A と省略することができます。
- **変更名** — 通り名が正式に変更された場合に、新しい名前を表すエイリアスです。例えば、SHINGLE BROOK RD という通り名が CANNING DR に変更された場合、CANNING DR が変更済みのエイリアス タイプとなります。
- **その他の名前** — この通り名エイリアスには、通りの他の名前や、通りの一般的な省略形などがあります。

エイリアスではない通り名のことを、基本通り名と呼びます。

入力において、よく使用されるエイリアスが使用されている場合は、このオプションを選択しているかどうかにかかわらず、そのエイリアスが出力の通り名になります。

これは、ValidateAddress でストリート名のエイリアスを処理する方法を制御する 3 つのオプションのうちの 1 つです。他の 2 つは Option.OutputStreetNameAlias と Option.OutputAbbreviatedAlias です。

多くの場合、Option.OutputPreferredAlias と Option.OutputAbbreviatedAlias の両方が選択されており、ValidateAddress が、郵便データベース内でよく使用されるエイリアスと省略形エイリアスの両方を検出した場合は、省略形エイリアスが出力に使用されます。入力の通り名がよく使用されるエイリアスである場合は、例外になります。この場合は、よく使用されるエイリアスが出力に使用されます。

- Y** 通り名に対してよく使用されるエイリアスの処理を実行します。
- N** 通り名に対してよく使用されるエイリアスの処理を実行しません。デフォルト

注: 入力住所に、"変更名"である通り名のエイリアスが含まれている場合は、指定したオプションにかかわらず、出力住所には必ず、基本通り名が使用されます。

パラメータ	説明
Option.OutputAbbreviatedAlias	<p>出力住所行の長さが 31 文字以上になる場合に、通り名に対する省略形エイリアスを出力に使用するかどうかを指定します。</p> <p>これは、<code>ValidateAddress</code> でストリート名のエイリアスを処理する方法を制御する 3 つのオプションのうちの 1 つです。他の 2 つは <code>Option.OutputStreetNameAlias</code> と <code>Option.OutputPreferredAlias</code> です。</p> <p>注: 入力において、よく使用されるエイリアスが指定されている場合は、通り名に対する省略形エイリアスの処理を有効にしても、出力の通り名は必ず、よく使用されるエイリアスになります。</p> <p><b>Y</b>      省略形エイリアスの処理を実行します。</p> <p><b>N</b>      省略形エイリアスの処理を実行しません。デフォルト</p> <p>注: 入力住所に、"変更名"である通り名のエイリアスが含まれている場合は、指定したオプションにかかわらず、出力住所には必ず、基本通り名が使用されます。</p>
Option.DPVDetermineNoStat	<p>住所の "no stat" ステータスを調べます。住所が存在するが、郵便物を受け取れない場合、その住所は "no stat" とみなされるため、配達ルートに関する配達統計としてカウントされません (そのため "no stat" という用語が使用されます)。例としては、建設中の建物や、郵便物を受け取る可能性が低いと郵便配達業者が識別した建物などがあります。</p> <p><b>N</b>      "no stat" ステータスを調べません。デフォルト</p> <p><b>Y</b>      "no stat" ステータスを調べます。</p> <p>注: このオプションを使用するには DPV 処理を有効にする必要があります。</p> <p>結果は <code>DPVNoStat</code> フィールドに返されます。詳細については、<a href="#">LACSLink 出力 (648ページ)</a></p>
Option.DPVDetermineVacancy	<p>そのロケーションがすくなくとも 90 日間使用されていないかどうかを調べます。</p> <p><b>N</b>      空家かどうかを調べません。デフォルト</p> <p><b>Y</b>      空家かどうかを調べます。</p> <p>注: このオプションを使用するには DPV 処理を有効にする必要があります。</p> <p>結果は <code>DPVVacant</code> フィールドに返されます。詳細については、<a href="#">LACSLink 出力 (648ページ)</a></p>

パラメータ	説明
Option.ReturnVerimove	<p>出力に VeriMove 詳細データを返します。</p> <p><b>N</b> VeriMove 詳細データを返しません。デフォルト</p> <p><b>Y</b> VeriMove 詳細データを返します。</p>
Option.SuppressZipPlusPhantomCarrierR777	<p>キャリアルート R777 の住所を抑制するかどうかを指定します。これらの住所は疑似ルートであり、通り配達に使用できません。これらの住所には USPS® による ZIP + 4® コードが割り当てられているため、Validate Address はこれらの住所を配達可能と判定します。キャリアルート R777 の住所を配達可能と判定したくない場合は、このオプションを選択します。その場合は、次のように動作します。</p> <ul style="list-style-type: none"> <li>• ZIP + 4 コードは割り当てられません</li> <li>• 住所は USPS Form 3553 (CASS Summary Report) から除外されます</li> <li>• DPV 補足コードとして R7 が返されます</li> </ul> <p><b>N</b> キャリアルート R777 の住所を抑制しません。</p> <p><b>Y</b> キャリアルート R777 の住所を抑制します。</p>
Option.StreetMatchingStrictness	<p>入力住所が郵便データベース内の住所にマッチするかどうかを調べる際に使用するアルゴリズムを指定します。次のいずれかです。</p> <p><b>E</b> 入力された通り名は、データベースに完全に一致する必要があります。</p> <p><b>T</b> マッチングアルゴリズムは "厳格" です。</p> <p><b>M</b> マッチングアルゴリズムは "中" です (デフォルト)。</p> <p><b>L</b> マッチングアルゴリズムは "あいまい" です。</p>
Option.FirmMatchingStrictness	<p>入力住所が郵便データベース内の住所にマッチするかどうかを調べる際に使用するアルゴリズムを指定します。次のいずれかです。</p> <p><b>E</b> 入力された企業名は、データベースに完全に一致する必要があります。</p> <p><b>T</b> マッチングアルゴリズムは "厳格" です。</p> <p><b>M</b> マッチングアルゴリズムは "中" です (デフォルト)。</p> <p><b>L</b> マッチングアルゴリズムは "あいまい" です。</p>

パラメータ	説明
Option.DirectionMatchingStrictness	<p>入力住所が郵便データベース内の住所にマッチするかどうかを調べる際に使用するアルゴリズムを指定します。次のいずれかです。</p> <p><b>E</b> 123 N Main St. における "N" など、入力された道順がデータベースに完全に一致する必要があります。</p> <p><b>T</b> マッチングアルゴリズムは "厳格" です。</p> <p><b>M</b> マッチングアルゴリズムは "中" です。デフォルト</p> <p><b>L</b> マッチングアルゴリズムは "あいまい" です。</p>
Option.DPVSuccessfulStatusCondition	<p>DPV 結果がレコードの失敗の原因とならない一致条件を選択します。</p> <p><b>F</b> 全体一致</p> <p><b>P</b> 部分一致</p> <p><b>A</b> 常に一致。デフォルト</p> <p>注：このオプションを使用するには DPV 処理を有効にする必要があります。</p>
Option.FailOnCMRAMatch	<p>民間私書箱（CMRA）との一致をマッチとみなしませんか。</p> <p><b>N</b> いいえ、CMRA との一致をマッチとみなします。デフォルト</p> <p><b>Y</b> はい、CMRA との一致をマッチとみなしません。</p> <p>注：このオプションを使用するには DPV 処理を有効にする必要があります。</p>
Option.StandardAddressPMBLine	<p>私書箱 (PMB) の情報をどこに配置するかを指定します。</p> <p><b>N</b> 標準住所出力に PMB 情報を含めません (デフォルト)。</p> <p><b>1</b> PMB 情報を AddressLine1 に配置します。1 を指定した場合、Option.StandardAddressFormat に C または D をセットする必要があります。</p> <p><b>2</b> PMB 情報を AddressLine2 に配置します。</p>

パラメータ	説明
Option.PreferredCity	<p>優先する最終行都市名を格納するかどうかを指定します。</p> <p><b>Z</b> USPS ZIP+4 ファイルからの Preferred Last Line City Name を格納します (都市名を上書き)。</p> <p>注：このオプションを選択すると、Validate Address は CASS 認定の設定と USPS 3553 レポートを生成します。</p> <p><b>C</b> USPS City/State ファイルからの USPS-preferred City Name を格納します</p> <p>注：このオプションを選択すると、Validate Address は CASS 認定の設定と USPS 3553 レポートを生成しません。</p> <p><b>P</b> USPS City/State ファイルからの Primary City Name を格納します</p> <p>注：このオプションを選択すると、Validate Address は CASS 認定の設定と USPS 3553 レポートを生成しません。</p>

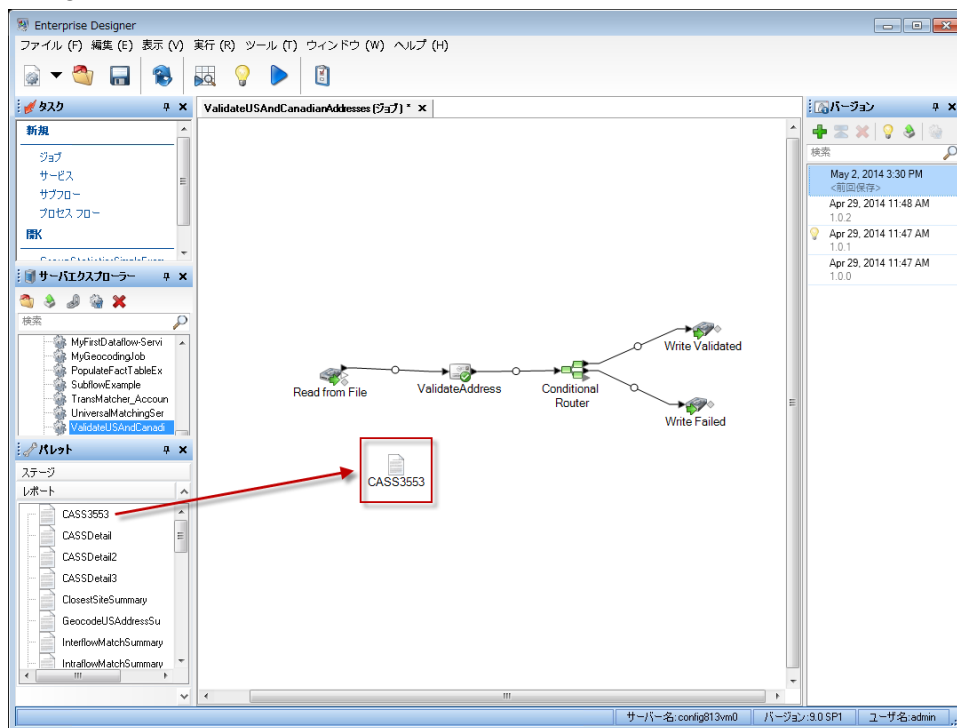
## CASS 認定処理

また、CASS 認定™処理では USPS CASS 詳細レポートも生成されます。このレポートに含まれる情報は 3553 レポートと同じものですが、DPV、LACS、および SuiteLink に関する大幅に詳しい統計情報が含まれます。USPS CASS 詳細レポートは、郵便料金の値引きを受けるために必ずしも必要ではなく、郵便物と一緒に提出する必要はありません。

1. Validate Address を CASS 認定™モードにする必要があります。ウィンドウの一番上に **(CASS 認定でないもの)** と表示される場合は、**[CASS 有効]** ボタンをクリックしてください。**[CASS ルールを強制]** チェック ボックスが表示されます。
2. **[CASS 3553 を設定]** をクリックします。**[CASS レポート フィールド]** ダイアログ ボックスが表示されます。
3. **[リスト処理元]** の会社名、**[リスト名または ID 番号]**、およびこのジョブで処理する **[リスト数]** を入力します。
4. **[差出人名]**、**[住所]**、および **[都市]**、**[州]**、**[ZIP]** を入力します。
5. **[OK]** をクリックします。

生成された USPS® CASS Form 3553 のセクション B にリストの情報が、セクション D に差出人の情報が表示されます。



6. Enterprise Designer で、**CASS3553** レポートをレポートのパレットからキャンバスにドラッグ

グします。

7. キャンバスの **[CASS3553]** アイコンをダブルクリックします。
8. **[ステージ]** タブで、**[Validate Address]** チェックボックスをオンにします。Validate Address ステージを何か別の名前に変更している場合は、住所検証ステージに指定した名前のチェックボックスをオンにする必要があります。
9. **[パラメータ]** タブで、レポートのフォーマットを選択します。PDF、HTML、またはプレーンテキストのフォーマットでレポートを作成できます。
10. **[OK]** をクリックします。
11. CASS 詳細レポートを生成する場合は、**CASSDetail** に対して手順 6 ~ 10 を繰り返します。

カナダ住所のオプション

## パラメータ

## 説明

---

**Option.PerformCanadianProcessing**

カナダ住所を処理するかどうかを指定します。カナダ住所処理を有効にした場合、**ValidateAddress** はカナダ住所の検証を試みます。カナダ住所処理を無効にすると、**Status** 出力フィールドに "F" が設定されて返され、カナダ住所処理は失敗します。その際、出力フィールド **Status.Code** は、"DisabledCoder" となります。カナダ住所処理のライセンスを取得していない場合は、ジョブにカナダ住所が含まれるか否かにかかわらず、カナダ住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：カナダ住所を正常に処理するには、カナダ住所処理の有効なライセンスを取得する必要があります。カナダ住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、カナダ住所処理を有効にすると、エラーとなります。

- N**      カナダ住所を処理しません。
- Y**      カナダ住所を処理します (デフォルト)。

---

**Option.Database.Canada**

カナダ住所の検証に使用するデータベースを指定します。カナダ住所検証用のデータベースを指定するには、**[データベース]** ドロップダウンリストからデータベースを選択します。**Management Console** の **[カナダ データベース リソース]** パネルで定義されたデータベースのみが使用可能です。

---

## パラメータ

## 説明

## Option.CanFrenchFormat

住所及び方向指示の書式設定に使用する言語 (英語またはフランス語) の判断方法を指定します。以下に、英語およびフランス語で書式設定された住所の例を示します。

英語: 123 Main St W フランス語: 123 Rue Main O

このパラメータは、住所の書式設定を制御します。また、方向指示の綴りにも影響を与えますが、接尾語の綴りには影響を与えません。

- C** マッチング処理によって返される通り接尾語によって、言語を判断します。マッチング処理によって返されるストリート接尾語は、`ValidateAddress` が処理において内部で使用するもので、入力住所のものとは異なる場合があります。あいまいなレコードは、入力と同様に書式設定されます。デフォルトケベック州の住所はすべて、フランス語で書式設定されます。
- S** カナダ データベースによって、言語を判断します。カナダ データベースには、**Canada Post Corporation (CPC: カナダ郵政公社)** からのデータが含まれています。ケベック州の住所はすべて、フランス語で書式設定されます。
- T** `CanLanguage` 入力フィールドによって、言語を判断します。このフィールドに空白以外の値が設定されている場合は、住所はフランス語で書式設定されます。

## Option.CanFrenchFormat

英語の住所に対し、入力住所にアパートメント ラベルが存在しない場合に、出力に使用するデフォルト アpartment ラベルを指定します。

`Option.CanStandardAddressFormat=F` と指定した場合、この設定は無視されます。

- Apt** ラベルとして "Apt" を使用します。デフォルト
- Apartment** ラベルとして "Apartment" を使用します。
- Suite** ラベルとして "Suite" を使用します。
- Unit** ラベルとして "Unit" を使用します。

## パラメータ

## 説明

## Option.CanFrenchApartmentLabel

フランス語の住所に対し、入力住所にアパートメント ラベルが存在しない場合に、出力に使用するデフォルト アパートメント ラベルを指定します。

Option.CanStandardAddressFormat=F と指定した場合、この設定は無視されます。

<b>App</b>	"App" をラベルとして使用します。デフォルト
<b>Appartement</b>	Use "Appartement" as the label.
<b>Bureau</b>	Use "Bureau" as the label.
<b>Suite</b>	Use "Suite" as the label.
<b>Unite</b>	Use "Unite" as the label.

## Option.ForceCorrectionLVR

正式情報やスイート情報を変更して、Large Volume Receiver (LVR) または Single-Single レコードをマッチさせます (その郵便番号/通り名/通り タイプに対して 1 つしかレコードがない場合に使用します)。

**N** LVR または Single-Single レコードをマッチさせるために正式情報やスイート情報を変更しません。LVR レコードは、有効だが修正不可能なレコード (VN) としてマーク付けされます。Single-Single レコードは可能ならば修正されます。または、修正不可能なレコードとして処理されます。

**Y** LVR または Single-Single レコードをマッチさせるために正式情報やスイート情報を変更します。

注：このチェック ボックスをオンにする場合は、SERP 認定の設定 ではないため、Statement of Address Accuracy は印刷されません。

パラメータ	説明
Option.CanPreferHouseNum	<p>家番号と郵便番号がともに有効であるが、競合する場合、Option.CanPreferHouseNum=Y と指定することによって、家番号に合わせて郵便番号を強制的に修正できます。このオプションを選択しない場合、郵便番号に合わせて家番号が変更されます。</p> <p><b>N</b> 郵便番号に合わせて家番号を変更します。デフォルト</p> <p><b>Y</b> 家番号に合わせて郵便番号を変更します。</p>
Option.CanOutputCityAlias	<p>入力住所に都市名のエイリアスがある場合、そのエイリアスを返すかどうかを指定します。Option.CanOutputCityFormat=D を指定している場合は、このオプションは無効です。</p> <p><b>Y</b> 入力に都市名のエイリアスがある場合は、都市名のエイリアスを出力します。デフォルト</p> <p><b>N</b> 入力に都市名のエイリアスがあっても、都市名のエイリアスを出力しません。</p>
Option.CanNonCivicFormat	<p>出力において、正式住所ではないキーワードを短縮するかどうかを指定します。例えば、Post Office Box と PO Box のどちらを使用するかが決まります。</p> <p><b>A</b> 正式住所ではないキーワードを短縮します。デフォルト</p> <p><b>F</b> 正式住所ではないキーワードを短縮しません。正式なキーワードを使用します。</p>
Option.EnableSERP	<p>SERP オプションを使用するかどうかを指定します。</p> <p><b>Y</b> SERP オプションを有効にします。</p> <p><b>N</b> SERP オプションを有効にしません。デフォルト</p>

## パラメータ

## 説明

## Option.CanStandardAddressFormat

出力住所における、補助的な住所情報の配置場所を指定します。補助的な住所情報とは、部屋番号やアパート番号などの指定子のことです。

- D** アパート情報を、で指定された場所に配置します。  
デフォルト
- B** アパート情報を、AddressLine1 フィールドの末尾に配置します。
- F** アパート番号のみ(ラベルは除く)を、AddressLine1 フィールドの先頭に配置します。例えば、400-123 Rue Main とします。
- E** アパート番号とラベルを、AddressLine1 フィールドの先頭に配置します。例えば、Apt 400 123 Rue Main とします。
- S** アパート情報を別の行に配置します。
- S** アパート情報を入力住所と同じ場所に配置します。

## Option.CanOutputCityFormat

都市の名前が長い場合に、long、medium、もしくはshortのどの都市名を使用するかを指定します。例を次に示します。

Long: BUFFALO HEAD PRAIRIE

Medium: BUFFALO-HEAD-PR

Short: BUFFALO-HD-PR

- D** Option.OutputShortCityNameパラメータで指定されたデフォルト オプションを使用します。デフォルト  
Option.OutputShortCityName=V を指定した場合、都市は、このオプションでを選択し(以下を参照)、CanOutputCityAlias で Y を選択した場合と同じように書式設定されます。
- S** 短い都市名を出力します。
- L** 長い都市名を出力します。
- M** 中間の長さの都市名を出力します。
- I** 入力住所と同じ都市フォーマットを使用します。出力は、L、M、または S です。

## パラメータ

## 説明

---

**Option.CanRuralRouteFormat**

地方配送路の配達情報を配置する場所を指定します。地方配送路の配達情報を含む住所の例を以下に示します。

**36 GRANT RD RR 3ANTIGONISH NS**

この住所において、"RR 3"は地方配送路の配達情報です。

- A** 地方配送路の配達情報を、住所と同一行の住所情報の後に配置します。デフォルト例を次に示します。

**36 GRANT RD RR 3**

- S** 地方配送路の配達情報を、別の住所行に配置します。例を次に示します。

**36 GRANT RDRR 3**

---

**Option.CanDeliveryOfficeFormat**

配達局情報の配置場所を指定します。配達局情報を含む住所の例を次に示します。

**PO BOX 8625 STN AST.JOHN'S NL**

- I** 配達局情報を、入力住所と同じ場所に配置します。デフォルト

- A** 配達局情報を、住所と同一行の住所情報の後に配置します。例を次に示します。

**PO BOX 8625 STN A**

- S** 配達局情報を、別の住所行に配置します。例を次に示します。

**PO BOX 8625STN A**

---

## パラメータ

## 説明

## Option.CanDualAddressLogic

住所に、正式情報と非正式情報の両方が含まれている場合に、**ValidateAddress** がストリート一致と、**PO Box**/非正式一致のどちらを返すかを指定します。次のいずれかです。

- D** **DualAddressLogic** のグローバル オプションを使用します。デフォルト
- P** **PO Box** など通り以外のデータとマッチングします。
- S** 通りとマッチングします。

例えば、次の入力住所が与えられたとします。

AddressLine1: 36 GRANT RD  
AddressLine2: RR 4  
City: ANTIGONISH  
StateProvince: NS

**ValidateAddress** は、次のいずれかを返します。

- **Option.CanDualAddressLogic** が **S** に設定されている場合、**ValidateAddress** は次を返します。

AddressLine1: 36 GRANT RD  
AddressLine2: RR 3  
City: ANTIGONISH  
StateProvince: NS  
PostalCode: B2G 2L1

- **Option.CanDualAddressLogic** が **P** に設定されている場合、**ValidateAddress** は次を返します。

AddressLine1: RR 4  
City: ANTIGONISH  
StateProvince: NS  
PostalCode: B2G 2L2

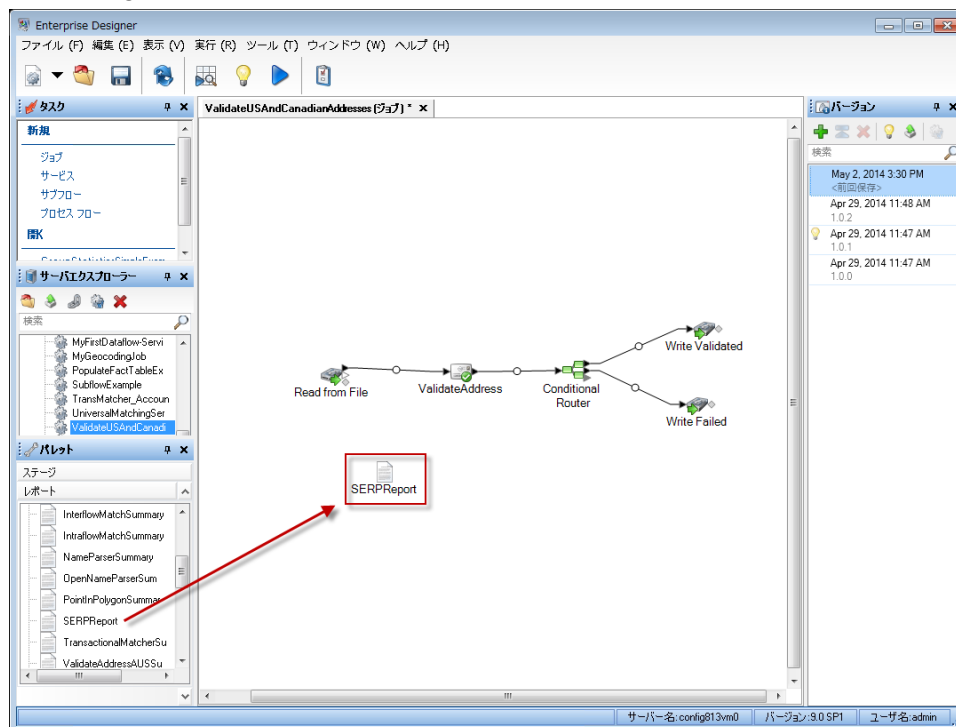
住所の正規化に使用されない住所データは、**AdditionalInputData** フィールドに返されます。詳細については、「[出力データオプション \(573ページ\)](#)」を参照してください。

## SERP の処理

1. **Validate Address** を SERP 認定™モードにする必要があります。ウィンドウの一番上に (SERP 認定でないもの) と表示される場合は、**[SERP 設定を有効にする]** ボタンをクリックしてください。**[SERP を設定]** ボックスが表示されます。
2. **[SERP を設定]** をクリックします。**[SERP レポート フィールド]** ダイアログ ボックスが表示されます。
3. 荷主の **[CPC 番号]** を入力します。
4. 差出人の **[名前]**、**[住所]**、および **[都市]**、**[州]**、**[郵便番号]** を入力します。



5. **[OK]** をクリックします。
6. Enterprise Designer で、SERP レポートをレポートのパレットからキャンバスにドラッグし



ます。

7. キャンバスの **[SERPReport]** アイコンをダブルクリックします。
8. **[ステージ]** タブで、**[Validate Address]** チェックボックスをオンにします。Validate Address ステージを何か別の名前に変更している場合は、住所検証ステージに指定した名前のチェックボックスをオンにする必要があります。
9. **[パラメータ]** タブで、レポートのフォーマットを選択します。PDF、HTML、またはプレーンテキストのフォーマットでレポートを作成できます。デフォルトではPDFフォーマットで作成されます。
10. **[OK]** をクリックします。

#### SERP リターン コードの取得

SERP リターン コードは、カナダ郵政公社の Software Evaluation and Recognition Program の規定によって定められる入力住所の品質を表します。

SERP リターン コードを取得するには、Option.OutputRecordType=P を指定します。Option.OutputRecordType の詳細については、[出力データ オプション](#) (573ページ) を参照してください。

SERP リターン コードは、以下の出力フィールドに返されます。

表 43 : SERP リターン コード出力

応答要素	説明
CanadianSERPCode	<p data-bbox="557 443 1032 474">検証/修正リターン コード (カナダ住所のみ)。</p> <p data-bbox="557 491 1403 558"><b>V</b> 入力は無効です。カナダ郵政公社は、以下のすべての条件を満たす住所を、「有効な」住所であると定義しています。</p> <p data-bbox="691 575 1403 642">注：一部例外があります。詳細については、CPC にお問い合わせください。</p> <ul data-bbox="607 680 1421 1050" style="list-style-type: none"> <li>• 住所は、CPC の Postal Code Data Files に示されるとおりに、すべての必須コンポーネントを含む必要があります。</li> <li>• 住所は、CPC の Postal Code Data Files の単一の住所のみに対し、すべてのコンポーネントにおいて完全に一致する必要があります。ただし、CPC Postal Code Data Files に示されている、許容される別の語および名前が使用されていてもかまいません。</li> <li>• 住所コンポーネントは、あいまいな部分がなく、はっきりと認識できる形式である必要があります。一部のコンポーネントには、それらを識別するための "修飾子" が必要な場合があります。例えば、Route Service の住所には、同じ番号の "Suburban Service" または "SS" の住所と区別するために、"Rural Route" または "RR" のキーワードが必要です。</li> </ul> <p data-bbox="557 1079 1421 1180"><b>I</b> 入力は無効です。「無効な」住所とは、有効な住所に対する CPC の条件を満たさない住所のことです (上記を参照)。例としては、住所コンポーネントが欠落している、無効である、または矛盾が存在する場合が挙げられます。</p> <p data-bbox="557 1205 1421 1272"><b>C</b> 入力は修正可能です。「修正可能な」住所とは、修正することによって、単一の住所のみに一致させることのできる住所のことです。</p> <p data-bbox="557 1289 1421 1390"><b>N</b> 入力は修正不可能です。「修正不可能な」住所とは、複数の異なる修正方法があり得るために ValidateAddress が単一の修正住所を特定できない住所のことです。</p> <p data-bbox="557 1415 1052 1440"><b>F</b> 入力住所は外国 (カナダ以外) の住所です。</p>

## 国際住所オプション

米国とカナダ以外の住所は "国際" 住所と呼ばれます。以下に、国際住所の処理をコントロールするオプションについて説明します。

## パラメータ

## 説明

---

**Option.PerformInternationalProcessing**

国際住所(米国およびカナダ以外の住所)を処理するかどうかを指定します。国際住所処理を有効にした場合、**ValidateAddress** は国際住所の妥当性を確認します。国際住所処理を無効にした場合、**Status**フィールドに "F" が設定され国際住所は失敗します。出力フィールド **Status.Code** は、"DisabledCoder" となります。国際住所処理のライセンスを取得していない場合は、ジョブに国際住所が含まれるか否かにかかわらず、国際住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：国際住所を正常に処理するには、国際住所処理の有効なライセンスを取得する必要があります。国際住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、国際住所処理を有効にすると、エラーとなります。

**N** 国際住所を処理しません。

**Y** 国際住所を処理します (デフォルト)。

---

**Option.Database.International**

国際住所の妥当性の確認に使用するデータベースを指定します。国際住所検証用のデータベースを指定するには、**【データベース】**ドロップダウンリストからデータベースを選択します。**Management Console** の **【国際データベースリソース】** パネルで定義されたデータベースのみが使用可能です。

## パラメータ

## 説明

---

**Option.InternationalCityStreetSearching**

デフォルトでは、**ValidateAddress** は、住所マッチングの精度とパフォーマンスのバランスをうまくとります。マッチング精度を犠牲にしてパフォーマンスを向上させる場合は、**Option.InternationalCityStreetSearching** パラメータを使用して、処理速度を上げます。これを実行すると、精度はやや低下します。このオプションは、米国およびカナダ以外の住所のパフォーマンスのみを制御します。この設定が影響を与えるレコードの割合は少なく、大部分が英国の住所です。米国およびカナダ住所処理のパフォーマンスは制御できません。

**GetCandidateAddresses** を使用した場合に

**GetCandidateAddresses** が返す候補住所は、国際住所のパフォーマンス チューニング オプションを 100 以外の任意の値に設定した場合に **ValidateAddress** が返す複数マッチと異なることがあります。

パフォーマンスを制御するには、0 ~ 100 の値を指定してください。100 を設定すると精度が最大化し、0 を設定すると速度が最大化します。デフォルト値は 100 です。

---

**Option.AddressLineSearchOnFail**

このオプションにより、**ValidateAddress** において、**City**、**StateProvince**、および **PostalCode** の各入力フィールドの値を使用して住所にマッチする結果が得られなかった場合に、**AddressLine** 入力フィールドで都市、州/省、郵便番号、および国を検索することができます。

入力住所において、**AddressLine** フィールドに都市、州/省、および郵便番号の情報が存在する場合は、このオプションを有効にすることを検討してください。

入力住所において、**City**、**State/Province**、および **PostalCode** フィールドが使用されている場合は、このオプションを無効にしてください。このオプションを有効にしてこれらのフィールドを使用すると、**ValidateAddress** がこれらのフィールド値の修正 (例えば、スペルミスのある都市名など) に失敗する可能性が高くなります。

**N** いいえ、**AddressLine** フィールドを検索しません。

**Y** はい、**AddressLine** フィールドを検索します。こちらがデフォルトです。

---

## 応答

`ValidateAddress` からの出力には、選択した出力カテゴリに応じて異なる情報が含まれます。

### 標準住所出力

標準住所出力は、宛名ラベルに表記される住所に対応する 4 行の住所で構成されます。都市、州/省、郵便番号などのデータも、標準住所出力に含まれます。`Option.OutputRecordType=A` と設定した場合、妥当性を確認した住所に対し、標準住所出力が返されます。妥当性が確認できなかった住所に対しては、標準住所フィールドが必ず返されます。妥当性が確認されなかった住所に対しては、標準住所出力フィールドには、入力住所がそのまま含まれます ("パス スルー" データ)。妥当性が確認できなかった場合に、郵便当局の規格に従って住所を正規化するには、リクエスト時に `Option.OutputFormattedOnFail=Y` を指定します。

表 44 : 標準住所出力

応答要素	説明
<code>AdditionalInputData</code>	住所検証プロセスで使用されない入力データ。詳細については、「 <a href="#">AdditionalInputData について (654 ページ)</a> 」を参照してください。
<code>AddressLine1</code>	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 1 行目です。住所の妥当性が確認できなかった場合は、入力住所の 1 行目がそのまま出力されます。
<code>AddressLine2</code>	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 2 行目です。住所の妥当性が確認できなかった場合は、入力住所の 2 行目がそのまま出力されます。
<code>AddressLine3</code>	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 3 行目です。住所の妥当性が確認できなかった場合は、入力住所の 3 行目がそのまま出力されます。
<code>AddressLine4</code>	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 4 行目です。住所の妥当性が確認できなかった場合は、入力住所の 4 行目がそのまま出力されます。

応答要素	説明
AddressLine5	英国住所にのみ適用されます。住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の5行目です。住所の妥当性が確認できなかった場合は、入力住所の5行目がそのまま出力されます。
City	妥当性が確認された都市名。
Country	Data.OutputCountryFormat で選択した、以下のいずれかのフォーマットで示された国。 <ul style="list-style-type: none"><li>• ISO コード</li><li>• UPU コード</li><li>• 英語</li><li>• フランス語</li><li>• ドイツ語</li><li>• スペイン語</li></ul>
DepartmentName	企業内の下位区分(英国住所においてのみ)。例えば、エンジニアリング部門などです。
FirmName	妥当性が確認された企業名。
PostalCode	妥当性が確認された ZIP Code™ または郵便番号。
PostalCode.AddOn	ZIP Code™ の4桁のアドオン部分。例えば、60655-1844 という ZIP Code™ において、4桁のアドオン部分は 1844 になります(米国住所のみ)。
PostalCode.Base	5桁の ZIP Code™。例えば、20706 (米国住所のみ)。
StateProvince	妥当性が確認された州または省の略称。
USUrbanName	妥当性が確認された都市の都市化名。(米国住所のみ)。主にプエルトリコの住所で使用されます。

## パース済み住所要素出力

Option.OutputRecordType=E を設定した場合、出力住所は、パース済み住所の形式で書式設定されます。妥当性が確認できなかった場合に、パース済み住所形式で書式設定されたデータ (正規化済み住所) を返すには、Option.OutputFormattedOnFail=Y を指定します。

注：妥当性が確認できたかどうかにかかわらず、常にパースした入力データを返すには、Option.OutputRecordType=I を指定します。詳細については、「[パース済み入力 \(700ページ\)](#)」を参照してください。

表 45 : パース済み住所出力

応答要素	説明
AdditionalInputData	ValidateAddress で使用されない入力データ。詳細については、「 <a href="#">AdditionalInputData について (654ページ)</a> 」を参照してください。
AdditionalInputData.Base	ValidateAddress によって正規化済み住所に出力されなかった入力データ。詳細については、「 <a href="#">AdditionalInputData について (654ページ)</a> 」を参照してください。
AdditionalInputData.Unmatched	マッチャーに引き渡されたが、ValidateAddress による検証に使用されなかった入力データ。詳細については、「 <a href="#">AdditionalInputData について (654ページ)</a> 」を参照してください。
ApartmentLabel	アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b>
ApartmentLabel2	補助的なアパート指定子。例: 123 E Main St Apt 3, 4th <b>Floor</b>  注: このリリースでは、このフィールドは常に空白になります。
ApartmentNumber	アパート番号。例: 123 E Main St <b>APT 3</b>

応答要素	説明
ApartmentNumber2	補助的なアパート番号。例: 123 E Main St APT 3, <b>4th</b> Floor 注: このリリースでは、このフィールドは常に空白になります。
CanadianDeliveryInstallationAreaName	配達施設名 (カナダ住所のみ)
CanadianDeliveryInstallationQualifierName	配達施設の修飾子 (カナダ住所のみ)
CanadianDeliveryInstallationType	配達施設の種類 (カナダ住所のみ)
City	妥当性が確認された都市名
Country	国。フォーマットは、 <code>Data.OutputCountryFormat</code> で選択したのになります。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> <li>• フランス語</li> <li>• ドイツ語</li> <li>• スペイン語</li> </ul>
DepartmentName	企業内の下位区分(英国住所においてのみ)。例えば、エンジニアリング部門などです。
FirmName	妥当性が確認された企業名
HouseNumber	家番号 1。例: <b>123</b> E Main St Apt 3
LeadingDirectional	接頭方向指示。例: <b>123 E</b> Main St Apt 3



応答要素	説明
POBox	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode	妥当性が検証された郵便番号。米国式の住所の場合に、ZIP Code と呼びます。
PrivateMailbox	私設私書箱のインジケータ。
PrivateMailbox.Type	私設私書箱の種別。次のような値があります。 <ul style="list-style-type: none"> <li>• Standard</li> <li>• Non-Standard</li> </ul> <p>注：PrivateMailboxType (フィールド名にピリオドなし) に換わるものです。これに従って API 呼び出しを修正してください。</p>
RRHC	地方配送路/幹線請負契約のインジケータ
StateProvince	妥当性が確認された州または省の名前
StreetName	通り名。例: 123 E <b>Main St</b> Apt 3
StreetSuffix	通り接尾語。例: 123 E Main <b>St</b> Apt 3
TrailingDirectional	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>
USUrbanName	USPS® 都市化名。プエルトリコ住所のみ。

### パース済み入力

出力には、パース済み形式で入力住所を含めることができます。このようなタイプの出力は、"パース済み入力" と呼ばれます。パース済み入力フィールドには、**ValidateAddress** が住所の妥当性を

検証したかどうかにかかわらず、入力として使用される住所データが含まれます。パース済み入力は、住所の妥当性を検証できた場合にパース済み住所要素に妥当性が検証された住所が含まれ、オプションで、住所の妥当性が検証できなかった場合には入力データが含まれるという点で、"パース済み住所要素" 出力と異なります。パース済み入力には、`ValidateAddress` が住所の妥当性を検証したかどうかにかかわらず、常に入力住所が含まれます。

パース済み入力フィールドを出力に含めるには、`Option.OutputRecordType=I` を設定します。

表 46 : パース済み入力

応答要素	説明
<code>ApartmentLabel.Input</code>	アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b>
<code>ApartmentNumber.Input</code>	アパート番号。例: 123 E Main St <b>APT 3</b>
<code>CanadianDeliveryInstallationAreaName.Input</code>	配達施設名 (カナダ住所のみ)
<code>CanadianDeliveryInstallationQualifierName.Input</code>	配達施設の修飾子 (カナダ住所のみ)
<code>CanadianDeliveryInstallationType.Input</code>	配達施設の種類 (カナダ住所のみ)
<code>City.Input</code>	妥当性が確認された都市名
<code>Country.Input</code>	国。フォーマットは、 <code>Data.OutputCountryFormat</code> で選択したものになります。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> <li>• フランス語</li> <li>• ドイツ語</li> <li>• スペイン語</li> </ul>

応答要素	説明
FirmName.Input	妥当性が確認された企業名
HouseNumber.Input	家番号 1。例: <b>123</b> E Main St Apt 3
LeadingDirectional.Input	接頭方向指示。例: 123 <b>E</b> Main St Apt 3
POBox.Input	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode.Input	妥当性が検証された郵便番号。米国住所に対しては、ZIP Code と呼びます。
PrivateMailbox.Input	私設私書箱のインジケータ
PrivateMailbox.Type.Input	私設私書箱の種別。次のような値があります。 <ul style="list-style-type: none"><li>標準</li><li>非標準</li></ul>
RRHC.Input	地方配送路/幹線請負契約のインジケータ
StateProvince.Input	妥当性が確認された州または省の名前
StreetName.Input	通り名。例: 123 <b>E Main St</b> Apt 3
StreetSuffix.Input	通り接尾語。例: 123 E Main St Apt 3
TrailingDirectional.Input	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>

応答要素	説明
------	----

USUrbanName.Input	USPS® 都市化名
-------------------	------------

### 郵便データ出力

Option.OutputRecordType に P が含まれる場合、以下のフィールドが出力として返されま  
す。

**表 47 : 郵便データ出力**

応答要素	説明
CanadianSERPCode	検証/修正リターン コード (カナダ住所のみ)。詳細については、「 <a href="#">SERP リターン コードの取得 (612ページ)</a> 」を参照してください。
IntHexaviaCode	通りを表す数値コード (フランスの住所の場合のみ)。Hexavia コードの詳細につい ては、 <a href="http://www.laposte.fr">www.laposte.fr</a> を参照してください。
IntINSEECODE	都市を表す数値コード (フランスの住所の場合のみ)。INSEE コードの一覧につい ては、 <a href="http://www.insee.fr">www.insee.fr</a> を参照してください。
PostalBarCode	配達ポイント バーコードの 2 桁の配達ポイント部分 (米国住所のみ)。詳細につい ては、「 <a href="#">配達ポイント バーコードの作成 (582ページ)</a> 」を参照してください。
USAltAddr	他の住所マッチング ロジックを使用したかどうか、使用した場合はどのロジック を使用したかを表します (米国住所のみ)。次のいずれかです。 <b>NULL</b> 他の住所スキームを使用していません。 <b>D</b> 別の配達ポイント ロジックを使用しました。 <b>E</b> 別の高層マッチ ロジックを使用しました。 <b>S</b> 小都市デフォルト ロジックを使用しました。 <b>U</b> ユニークな ZIP Code ロジックを使用しました。

応答要素	説明
USBCCheckDigit	11桁の配達ポイントバーコードのチェック デジット部分 (米国住所のみ)。詳細については、「 <a href="#">配達ポイントバーコードの作成 (582ページ)</a> 」を参照してください。
USCarrierRouteCode	配達ルート コード (米国住所のみ)。詳細については、「 <a href="#">配達ルート コードの取得 (582ページ)</a> 」を参照してください。
USCongressionalDistrict	下院選挙区出力 (米国住所のみ)。詳細については、「 <a href="#">下院選挙区の取得 (580ページ)</a> 」を参照してください。
USCountyName	郡名出力 (米国住所のみ)。詳細については、「 <a href="#">郡名の取得 (581ページ)</a> 」を参照してください。
USFinanceNumber	住所の所在地の Finance Number (米国住所のみ)。Finance Number とは、複数の ZIP Code を含む地域に USPS が割り当てた番号です。住所の Finance Number が米国データベースの候補住所の Finance Number に一致した場合のみ、住所の妥当性確認に成功します。
USFIPSCountyNumber	FIPS (連邦情報処理標準) 郡番号 (米国住所のみ)。詳細については、「 <a href="#">FIPS 郡番号の取得 (581ページ)</a> 」を参照してください。

## 応答要素

## 説明

## USLACS

住所が、LACS<sup>Link</sup> 変換の候補であるかどうかを表します (米国住所のみ)。次のいずれかです。

- Y** 住所は LACS<sup>Link</sup> 処理の候補です。LACS<sup>Link</sup> が有効である場合、LACS<sup>Link</sup> データベースを使用して住所を変換しようとします。変換に成功した場合、出力住所は LACS<sup>Link</sup> データベースから取得した新しい住所になります。変換できなかった場合は、住所は変換されません。
- N** 住所は LACS<sup>Link</sup> 処理の候補ではありません。ただし、LACS<sup>Link</sup> 処理が要求され、LACS<sup>Link</sup> データベースがインストールされており、かつ、次の条件のいずれかが満たされている場合は、LACS<sup>Link</sup> 処理が行われる場合があります。
  - 住所が地方配送路住所にマッチし、RecordType.Default フィールドで Y が返された場合。
  - 入力住所が、米国郵便データベースのいずれの住所にもマッチしなかった場合 (複数にマッチしたことによる失敗は、LACS<sup>Link</sup> の候補にはなりません)。

## USLastLineNumber

主要都市が同一である複数の ZIP Code を同一グループにまとめる 6 文字の英数字の値。例えば、最終行が次の 2 つのいずれかである住所は、最終行番号が同一になります。

Chantilly VA 20151

Chantilly VA 20152

## 結果インジケータ

結果インジケータは、住所に対して実行した処理の種類に関する情報を提供します。結果インジケータには、次の 2 種類があります。

## レコード レベルの結果インジケータ

レコード レベルの結果インジケータは、各レコードに対する ValidateAddress 処理の結果に関するデータを提供します。例えば、マッチングの成功または失敗、住所を処理したコーダーなどの詳細情報を示します。以下の表に、ValidateAddress が返すレコード レベルの結果インジケータの一覧を示します。

表 48 : レコード レベル インジケータ

応答要素	説明
AddressFormat	返された住所データのタイプ。 <b>F</b> フランス語フォーマット (例: 123 Rue Main) <b>E</b> 英語フォーマット (例: 123 Main St)
Confidence	返された住所に割り当てられた確信レベル。範囲は 0 ~ 100 です。0 は失敗を表し、100 はマッチ結果が正しいことに対する確信レベルが非常に高いことを表します。複数のマッチ結果がある場合、確信レベルは 0 です。この数値の計算方法については、 <a href="#">Validate Address 確信アルゴリズムの概要 (762ページ)</a> を参照してください。
CouldNotValidate	マッチしなかった場合の、妥当性が確認できなかった住所コンポーネント。 <ul style="list-style-type: none"> <li>• ApartmentNumber</li> <li>• HouseNumber</li> <li>• StreetName</li> <li>• PostalCode</li> <li>• City</li> <li>• Directional</li> <li>• StreetSuffix</li> <li>• Firm</li> <li>• POBoxNumber</li> <li>• RuralRoute</li> </ul> <p>注：複数のコンポーネントがカンマ区切りリストとして返されることがあります。</p>

## 応答要素

## 説明

## CountryLevel

利用可能な住所マッチングのカテゴリ。米国およびカナダの住所に対しては、常に「A」です。次のいずれかです。

- A** 住所は非常に詳細な郵便データを利用できる国にあります。このマッチレベルにある住所では、以下の住所要素を検証および修正でき、入力から欠落している場合は追加できます。
- 郵便番号
  - 都市名
  - 州/郡名
  - 通り住所要素
  - 国名
- B** 住所は中程度の詳細さの郵便データを利用できる国にあります。このマッチレベルにある住所では、以下の住所要素を検証および修正でき、入力から欠落している場合は追加できます。
- 郵便番号
  - 都市名
  - 州/郡名
  - 国名
- C** 住所は郵便データが詳細ではない国にあります。このマッチレベルにある住所に対して、以下のアクションを実行することができます。
- 国名の検証および修正 (欠落している国名を補うことはできません)
  - 郵便番号のフォーマットの検証 (欠落している郵便番号を補ったり、番号を検証することはできません)



## 応答要素

## 説明

---

**MatchScore**

**MatchScore** は、出力住所がどの程度正しいかを示します。**MatchScore** は、マッチ結果を得るために入力住所をどれだけ変更したかを表す **Confidence** とはまったく異なるものです。**MatchScore** の意味は、米国住所と米国以外の住所で異なります。

米国住所に対しては、**MatchScore** は 0 ~ 9 の段階に対応する 1 桁のスコアで、(**ValidateAddress** による変換があればその実行後に) 通り名一致の近接性を反映します。0 は完全一致を意味し、9 は最も可能性の低い一致を意味します。マッチしなかった場合、このフィールドは空白です。

米国とカナダ以外の住所では、**MatchScore** は 5 桁のスコアで、最大値は 00999 です。数字が大きいほど、より近い一致を意味します。

このフィールドは、カナダの住所には適用されません。

米国住所のマッチ スコアと米国以外の住所のマッチ スコアは、同等と見なすことはできないことに注意してください。例えば、米国住所に対するマッチスコア 4 は、米国以外の住所に対する 00004 と同じマッチ レベルを意味するものではありません。

注： **Validate Address** および **Advanced Matching** モジュールのコンポーネントは、どちらも **MatchScore** フィールドを使用します。データフローの出力の **MatchScore** フィールドの値は、出力ステージに送られる前に最後に値を変更したステージによって決まります。データフローに **Validate Address** および **Advanced Matching** モジュールのコンポーネントが含まれ、各ステージの **MatchScore** 出力フィールドを確認したい場合は、**Transformer** ステージを使用して、**MatchScore** 値を他のフィールドにコピーしてください。例えば、**Validate Address** によって **MatchScore** という出力フィールドが作成され、**Transformer** ステージによって **Validate Address** の **MatchScore** フィールドが **AddressMatchScore** というフィールドにコピーされます。マッチャー ステージを実行すると、マッチャーから得た値が **MatchScore** フィールドに設定され、**Validate Address** から得た **AddressMatchScore** の値が引き渡されます。

---

**MultimatchCount**

複数のマッチが検出された場合、一致する可能性のあるレコードの数を示します。

---

応答要素	説明						
MultipleMatches	<p>複数のマッチが検出された場合に、複数のマッチを持つ次の住所コンポーネントを示します。</p> <ul style="list-style-type: none"><li>• Firm</li><li>• LeadingDirectional</li><li>• PostalCode</li><li>• StreetName</li><li>• StreetSuffix</li><li>• TrailingDirectional</li><li>• Urbanization</li></ul> <p>注：複数のコンポーネントがカンマ区切りリストとして返されることがあります。</p>						
ProbableCorrectness	<p>一致の精度を 0 ~ 9 の段階で示します。結果は次のようになります。</p> <ul style="list-style-type: none"><li>• 空白 - 一致するものが見つかりませんでした</li><li>• 0 - 正確である可能性が非常に高い (完全一致)</li><li>• 1-8 - 正確である可能性は中程度</li><li>• 9 - 正確である可能性が非常に低い一致</li></ul>						
ProcessedBy	<p>住所を処理した住所コーダーです。</p> <table><tbody><tr><td><b>USA</b></td><td>米国住所コーダー</td></tr><tr><td><b>CAN</b></td><td>カナダ住所コーダー</td></tr><tr><td><b>INT</b></td><td>国際住所コーダー</td></tr></tbody></table>	<b>USA</b>	米国住所コーダー	<b>CAN</b>	カナダ住所コーダー	<b>INT</b>	国際住所コーダー
<b>USA</b>	米国住所コーダー						
<b>CAN</b>	カナダ住所コーダー						
<b>INT</b>	国際住所コーダー						
RecordType	<p>米国およびカナダの郵政当局が定義した住所レコードのタイプ (米国およびカナダの住所のみサポート):</p> <ul style="list-style-type: none"><li>• FirmRecord</li><li>• GeneralDelivery</li><li>• HighRise</li><li>• PostOfficeBox</li><li>• RRHighwayContract</li><li>• Normal</li></ul>						

応答要素	説明
RecordType.Default	<p>"デフォルト" マッチを示すコード</p> <p><b>Y</b> 住所はデフォルト レコードにマッチしています。</p> <p><b>NULL</b> 住所はデフォルト レコードにマッチしていません。</p>
Status	<p>マッチの成功または失敗。複数のマッチがある場合、一致する可能性のあるすべてのものに対してこのフィールドが "F" になります。</p> <p><b>NULL</b> 成功</p> <p><b>F</b> 失敗</p>
Status.Code	<p>失敗の原因 (ある場合)。複数のマッチがある場合、一致する可能性のあるすべてのものが "MultipleMatchesFound" になります。</p> <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• InsufficientInputData</li> <li>• MultipleMatchesFound</li> <li>• UnableToValidate</li> </ul>
Status.Description	<p>問題の説明 (ある場合)。</p> <p><b>Possible Multiple Addresses Found</b> Status.Code=MultipleMatchesFound の場合にこの値が表示されます。</p> <p><b>Address Not Found</b> Status.Code=UnableToValidate の場合にこの値が表示されます。</p> <p><b>PerformUSProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。</p> <p><b>PerformCanadianProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。</p> <p><b>PerformInternationalProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。</p>

## フィールドレベルの結果インジケータ

フィールドレベルの結果インジケータは、**ValidateAddress** が各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、

HouseNumber のフィールドレベルの結果インジケータは **HouseNumber.Result** に格納されます。

フィールドレベルの結果インジケータを有効にするには、Option.OutputFieldLevelReturnCodes=Y を指定します。詳細については、[出力データ オプション \(573ページ\)](#) を参照してください。

次の表に、フィールドレベルの結果インジケータの一覧を示します。特定のフィールドが住所に適用されない場合、結果インジケータが空白になる場合があります。

**表 49 : フィールドレベルの結果インジケータ**

応答要素	説明
AddressRecord.Result	<p>これらの結果コードは国際住所のみに適用されます。</p> <ul style="list-style-type: none"> <li data-bbox="816 873 1421 972"><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。</li> <li data-bbox="816 995 1421 1058"><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</li> <li data-bbox="816 1081 1029 1108"><b>U</b> マッチしない。</li> <li data-bbox="816 1131 1421 1199"><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</li> </ul>

## 応答要素

## 説明

---

**ApartmentLabel.Result**

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
  - C** 修正済み。米国およびカナダの住所のみをサポートします。
  - D** ドロップ済み入力に与えられたフィールドが削除されました。米国およびカナダの住所のみをサポートします。詳細については、「[その他の入力データについて](#)」を参照してください。
  - F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
  - P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
  - R** アパートラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
  - S** 正規化。このオプションには、標準の略語が含まれます。
  - U** マッチしない。カナダの住所には適用されません。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
-

## 応答要素

## 説明

## ApartmentNumber.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。カナダの住所のみ。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。EWS にマッチする米国の住所には、P の値が割り当てられます。米国およびカナダの住所のみをサポートします。
- R** アパート番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

## City.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- F** ハイフンの欠落または句読文字エラー。カナダの住所のみ。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。
- R** 都市名が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## Country.Result

これらの結果コードは、米国またはカナダの住所には適用されません。

- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

## FirmName.Result

- C** 修正済み。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
- U** マッチしない。米国およびカナダの住所のみをサポートします。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。米国住所にのみ適用されます。

## HouseNumber.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
- C** 修正済み。カナダの住所のみ。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- O** 範囲外。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
- R** 家番号が必須ですが、入力住所から欠落しています。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。



## 応答要素

## 説明

## LeadingDirectional.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。空白のない入力、空白のない値に修正されました。米国住所にのみ適用されます。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。カナダの住所には適用されません。

## 応答要素

## 説明

POBox.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
  - C** 修正済み。カナダの住所のみ。
  - D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
  - F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
  - M** 複数マッチ。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
  - P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。カナダの住所のみ。
  - R** 私書箱番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
  - S** 正規化。このオプションには、標準の略語が含まれます。
  - U** マッチしない。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
- 空白 不適用。

## 応答要素

## 説明

---

PostalCode.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
  - C** 修正済み。米国およびカナダの住所のみをサポートします。
  - F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
  - M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。カナダの住所には適用されません。
  - P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国住所には適用されません。
  - R** 郵便番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
  - S** 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。
  - U** マッチしない。例えば、通り名と郵便番号が一致しない場合、**StreetName.Result** と **PostalCode.Result** の両方に **U** が割り当てられます。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
-

## 応答要素

## 説明

## PostalCodeCity.Result

これらの結果コードは国際住所のみに適用されます。

- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## PostalCode.Source

これらの結果コードは米国住所にのみ適用されます。

- FinanceNumber** 入力の ZIP Code™ は、USPS® Finance Number グループを使って検証されました。
- ZIPMOVE** 入力住所の ZIP Code™ は、USPS® が改訂した ZIP Code™ 境界に基づいて修正され、住所に別の ZIP Code™ が設定されました。

## PostalCode.Type

- P** ZIP Code™ には、PO Box 住所のみが含まれます。米国住所にのみ適用されます。
- U** ZIP Code™ は、特定の会社または場所に割り当てられたユニークな ZIP Code™ です。米国住所にのみ適用されます。
- M** ZIP Code™ は、軍施設の住所です。米国住所にのみ適用されます。
- NULL** ZIP Code™ は、標準 ZIP Code™ です。

## 応答要素

## 説明

## RRHC.Result

- C** 修正済み。カナダの住所のみ。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- M** 複数マッチ。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。カナダの住所のみ。
- R** 地方配送路/幹線請負契約が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国およびカナダの住所のみをサポートします。
- U** マッチしない。米国およびカナダの住所のみをサポートします。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。米国およびカナダの住所のみをサポートします。

## RRHC.Type

これらの結果コードは米国住所にのみ適用されます。

- HC** 住所は、幹線請負契約住所です。
- RR** 住所は、地方配送路住所です。

## 応答要素

## 説明

## StateProvince.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国住所にのみ適用されます。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。
- R** アパート ラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## Street.Result

これらの結果コードは国際住所のみに適用されます。

- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。
- R** 通りが修正済みです。家番号が範囲外にあります。フランス、英国、および日本のレコードのみに適用。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

---

StreetName.AbbreviatedAlias.Result

省略形エイリアス処理の結果を示します。次のいずれかです。

- NULL** 省略形エイリアス処理が実行されませんでした。
  - B** StreetName フィールドに基本通り名が格納されています。
  - L** 正規化された住所長が 31 文字未満なので、StreetName フィールドに基本名が格納されています。
  - N** 省略形エイリアスが見つかりませんでした。
  - Y** 省略形エイリアスが入力住所に見つかりました。StreetName フィールドに省略形エイリアスが格納されています。
-

## 応答要素

## 説明

---

StreetName.Alias.Type

この結果コードは米国住所にのみ適用されます。

注：以前のリリースでは、このフィールドは "Alias" と "Type" の間に "." がない StreetName.AliasType という名前でした。この古い名前は廃止されました。新しい名前 StreetName.Alias.Type を使用するよう、プロセスを更新してください。

- Abbreviated** エイリアスは通り名の省略形です。例えば、HARTS-NM RD は HARTSVILLE NEW MARLBORO RD の省略形エイリアスです。
- Changed** 通り名が正式に変更された場合に、新しい名前を表すエイリアスです。例えば、SHINGLE BROOK RD という通り名が CANNING DR に変更された場合、CANNING DR が変更済みのエイリアス タイプとなります。
- Other** この通り名エイリアスには、通りの他の名前や、通りの一般的な省略形などがあります。
- Preferred** 通り名エイリアスはその地域でよく使用されるエイリアスです。例えば、ある通りが "South Shore Dr." という名前なのは、湖の南岸を通っているためで、地方自治体の境界線の南にあるからではありません。この場合、"South" は前置方位記号ではないので、"S" と短縮してはいけません。したがって、"South Shore Dr." がよく使用されるエイリアスになります。
-



## 応答要素

## 説明

---

StreetName.PreferredAlias.Result

よく使用されるエイリアス処理の結果を示します。次のいずれかです。

- NULL** よく使用されるエイリアス処理が実行されませんでした。
  - A** 入力住所がエイリアスにマッチしたため、よく使用されるエイリアス処理が実行されませんでした。よく使用されるエイリアス処理は、基本住所に対してのみ実行されます。
  - N** よく使用されるエイリアスが見つかりませんでした。
  - Y** 入力住所に対してよく使用されるエイリアスが見つかりました。**StreetName** フィールドによく使用されるエイリアスが格納されています。
-

## 応答要素

## 説明

---

**StreetName.Result**

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
  - C** 修正済み。米国およびカナダの住所のみをサポートします。
  - D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
  - F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
  - M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
  - P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国住所には適用されません。
  - S** 正規化。このオプションには、標準の略語が含まれます。米国およびカナダの住所のみをサポートします。
  - U** マッチしない。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
-

## 応答要素

## 説明

## StreetSuffix.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国およびカナダの住所のみをサポートします。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。米国住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

---

**TrailingDirectional.Result**

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国およびカナダの住所のみをサポートします。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

---

**USUrbanName.Result**

これらの結果コードは米国住所にのみ適用されます。

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。
  - C** 修正済み。
  - M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
  - U** マッチしない。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
-

## オプションによる出力

ValidateAddress は、選択したオプションに応じて、追加のデータを返します。各オプションによって生成される出力については、以下のセクションのオプションの一覧を参照してください。

### Enhanced Line of Travel 出力

Enhanced Line of Travel 処理では、以下の出力を生成します。

応答要素	説明
USLOTCode	Line of Travel の順序コードと、USPS® LOT 順序を表すインジケータ。このフィールドは、"nnnnY" という形式で、以下の要素で構成されます。 <ul style="list-style-type: none"> <li><b>nnnn</b>      4 桁の LOT コード。</li> <li><b>Y</b>            次のいずれかです。               <ul style="list-style-type: none"> <li>• <b>A</b> — 昇順の LOT 順序</li> <li>• <b>D</b> — 降順の LOT 順序</li> </ul> </li> </ul>
USLOTHex	昇順でのみファイルのソートが可能な 16 進値です。16 進値の範囲は、昇順で 0 ~ FF に続き、降順で FF ~ 0 です。
USLOTSequence	アドオンの DPC の代わりに最終仕分けに使用される 2 バイトの値。大文字 1 文字の後に 0 ~ 9 の数字が 1 つ続きます。値の範囲は、A0 (99 降順) ~ J9 (00 降順)、および K0 (00 昇順) ~ T9 (99 昇順) です。

LACS<sup>Link</sup> 出力

応答要素	説明
USLACS	<p>住所が、LACS<sup>Link</sup> 変換の候補であるかどうかを表します (米国住所のみ)。次のいずれかです。</p> <p><b>Y</b> 住所は LACS<sup>Link</sup> 処理の候補です。LACS<sup>Link</sup> が有効である場合は、ValidateAddress は、LACS<sup>Link</sup> データベースを使用して住所を変換します。変換に成功した場合、出力住所は LACS<sup>Link</sup> データベースから取得した新しい住所になります。変換できなかった場合は、住所は変換されません。</p> <p><b>N</b> 住所は LACS<sup>Link</sup> 処理の候補ではありません。ただし、LACS<sup>Link</sup> 処理が要求され、LACS<sup>Link</sup> データベースがインストールされており、かつ、次の条件のいずれかが満たされている場合は、LACS<sup>Link</sup> 処理が行われる場合があります。</p> <ul style="list-style-type: none"> <li>住所が地方配送路住所にマッチし、RecordType.Default フィールドで Y が返された場合。</li> <li>入力住所が、米国郵便データベースのいずれの住所にもマッチしなかった場合 (複数にマッチしたことによる失敗は、LACS<sup>Link</sup> の候補にはなりません)。</li> </ul>
USLACS.ReturnCode	<p>LACS<sup>Link</sup> 処理の成功または失敗を表します(米国住所のみ)。</p> <p><b>A</b> LACS<sup>Link</sup> 処理は成功しました。LACS<sup>Link</sup> 処理によってレコードがマッチしました。</p> <p><b>00</b> LACS<sup>Link</sup> 処理は失敗しました。LACS<sup>Link</sup> 処理において、マッチするレコードは見つかりませんでした。</p> <p><b>09</b> LACS<sup>Link</sup> 処理において、入力住所は、古い高層のデフォルト住所にマッチしました。住所は変換されています。不明確な住所の提供を避け、LACS<sup>Link</sup> 処理では、新しい住所を提供しません。</p> <p><b>14</b> LACS<sup>Link</sup> 処理は失敗しました。LACS<sup>Link</sup> 処理において、マッチする結果が検出されましたが、他の USPS<sup>®</sup> の規則に基づき、変換は行われませんでした。</p> <p><b>92</b> LACS<sup>Link</sup> 処理は成功しました。LACS<sup>Link</sup> 処理によってレコードがマッチしました。入力のユニット番号はドロップされました。</p> <p><b>NULL</b> LACS<sup>Link</sup> はレコードを処理しなかったか、または LACS<sup>Link</sup> 処理が実行されませんでした。</p>

## RDI 出力

応答要素	説明
RDI	住所の種類を表す値を返します。 <b>B</b> 住所は、企業住所です。 <b>R</b> 住所は、個人住所です。 <b>M</b> 住所は、個人住所であるとともに企業住所でもあります。 <b>NULL</b> 住所が ZIP + 4 <sup>®</sup> レベルでコード化されなかったか、または RDI™ が実行されなかったため、確認されていません。

## DPV および CMRA 出力

応答要素	説明
DPV	<p data-bbox="553 472 1138 499">Delivery Point Validation (DPV) 処理の結果を表します。</p> <p data-bbox="553 520 1247 548"><b>Y</b> DPV の確認済みです。この住所に郵便物を配達できます。</p> <p data-bbox="553 569 1024 596"><b>N</b> この住所に郵便物を配達できません。</p> <p data-bbox="553 617 1425 758"><b>S</b> 建物番号の妥当性は確認できましたが、ユニット番号は確認できませんでした。建物番号は、建物の主要な住所番号です。ユニット番号は、建物内のアパート、スイート、階など、各ユニットの郵便住所番号です。例えば、以下の住所の場合、<b>424</b> は建物番号、<b>12</b> はユニット番号です。</p> <p data-bbox="630 779 935 873">424 Washington Blvd.Apt.12 Oak Park IL 60302 USA</p> <p data-bbox="553 905 1425 1045"><b>D</b> 建物番号の妥当性は確認できましたが、ユニット番号は入力から欠落していました。建物番号は、建物の主要な住所番号です。ユニット番号は、建物内のアパート、スイート、階など、各ユニットの郵便住所番号です。例えば、以下の住所の場合、<b>424</b> は建物番号、<b>12</b> はユニット番号です。</p> <p data-bbox="630 1066 935 1161">424 Washington Blvd.Apt.12 Oak Park IL 60302 USA</p> <p data-bbox="553 1192 1170 1220"><b>M</b> 住所は複数の有効な配達ポイントにマッチします。</p> <p data-bbox="553 1241 1425 1304"><b>U</b> 住所は、ZIP + 4<sup>®</sup> レベルでコード化されなかったため、確認できませんでした。</p> <p data-bbox="553 1325 1024 1352"><b>V</b> 住所は、誤検出違反を起こしました。</p>
CMRA	<p data-bbox="553 1444 1425 1507">住所が Commercial Mail Receiving Agency (CMRA: 民間私書箱) であることを表します。</p> <p data-bbox="553 1528 922 1556"><b>Y</b> 住所は CMRA です。</p> <p data-bbox="553 1577 1040 1604"><b>N</b> 住所は CMRA ではありません。</p> <p data-bbox="553 1625 889 1652"><b>U</b> 確認できません。</p>



## 応答要素

## 説明

## DPVFootnote

DPV 補足コード。

- AA** 入力住所は、ZIP + 4<sup>®</sup> ファイルにマッチしました。
- A1** 入力住所は、ZIP + 4<sup>®</sup> ファイルにマッチしませんでした。
- BB** 入力住所は、DPV にマッチしました (すべてのコンポーネント)。
- CC** 入力住所の主要な番号は DPV にマッチしましたが、補助的な番号はマッチしませんでした (存在しましたが有効ではありませんでした)。
- F1** 入力住所は軍関係の住所。DPV は省かれます。
- G1** 入力住所は一般的な配達住所。DPV は省かれます。
- M1** 入力住所の主要な番号が欠落しています。
- M3** 入力住所の主要な番号が無効です。
- N1** 入力住所の主要な番号は DPV にマッチしましたが、高層住所に補助的な番号が欠落しています。
- P1** 入力住所に、RR または HC Box 番号がありません。
- P3** 入力住所に、PO、RR、または HC Box 番号がありません。
- RR** 入力住所は、CMRA にマッチしました。
- R1** 入力住所は CMRA にマッチしましたが、補助的な番号が存在しません。
- R7** 入力住所は擬似キャリアルート R777 にマッチしました (通り配達の対象になりません)。
- U1** 入力住所はユニーク ZIP。DPV は省かれます。

## DPVvacant

建物が空家 (90 日間使用されていない)かどうかを表します。次のいずれかです。

- Y** 建物は空家です。
- N** 建物は空家ではありません。
- NULL** [Option.DPVDetermineVacancy] オプションが選択されていません。

## 応答要素

## 説明

DPVNoStat

建物が、郵便物を受け取ることのできない "no stat" の建物であるかどうかを表します。次のいずれかです。

- Y** 建物は、郵便物を受け取ることのできない "no stat" の建物です。
- N** 建物は、郵便物を受け取ることのできない "no stat" の建物ではありません。
- NULL** オプションが選択されていません。

Suite<sup>Link</sup> 出力

## 応答要素

## 説明

SuiteLinkReturnCode

ValidateAddress が、補助的な住所情報を修正したかどうかを表します (米国住所のみ)。次のいずれかです。

- A** ValidateAddress は、補助的な住所情報を修正しました。
- 00** ValidateAddress は、補助的な住所情報を修正しませんでした。
- NULL** Suite<sup>Link</sup> は実行されませんでした。
- XX** Suite<sup>Link</sup> 処理においてエラーが発生しました。例えば、Suite<sup>Link</sup> データベースの有効期限が切れている場合にエラーが発生します。

## 応答要素

## 説明

## SuiteLinkMatchCode

Suite<sup>Link</sup> のマッチング処理に関する追加情報を提供します(米国住所のみ)

- A** ValidateAddress は、補助的な住所情報を修正しました。
- B** ValidateAddress は、補助的な住所情報を修正しませんでした。マッチング処理に関するその他の詳細情報はありません。
- C** FirmName フィールドの語はすべて、「ノイズ」語です。ノイズ語は、USPS<sup>®</sup> によって定義されており、企業名のマッチングの際には無視されます。ノイズ語の例としては、"company" や "corporation" があります。ValidateAddress は、ノイズ語のみで構成される企業名に対し、補助的な住所情報を修正できません。例えば、"Company and Corporation" という表記は、ノイズ語のみで構成されています。
- D** 住所は、高層のデフォルト住所ではありません。Suite<sup>Link</sup> マッチングは、高層のデフォルト住所に対してのみ行われます。高層デフォルトとは、住所に有効な補助的な情報が含まれていない(アパート番号やアパート種別が欠落している)場合に使用されるデフォルトです。
- E** Suite<sup>Link</sup> データベースの有効期限が切れているため、Suite<sup>Link</sup> 処理は失敗しました。
- NULL** Suite<sup>Link</sup> は実行されなかったか、エラーが発生しました。

## SuiteLinkFidelity

ValidateAddress における、Suite<sup>Link</sup> データベースの企業名に対するマッチング精度を表します。

- 1** 企業名は、Suite<sup>Link</sup> データベースに完全に一致しました。
- 2** 精度の高いマッチです。企業名に含まれる語が、1 語を除いてすべて Suite<sup>Link</sup> データベースの企業名に一致しました。
- 3** 精度の低いマッチです。企業名の中の複数の語が、Suite<sup>Link</sup> データベースの企業名に一致しませんでした。
- NULL** Suite<sup>Link</sup> が企業名のマッチングに失敗したか、実行されなかったか、またはエラーが発生しました。

## VeriMove 出力

応答要素	説明
VeriMoveDataBlock	<p>ValidateAddress が、VeriMove Express に渡される入力データを含む 250 バイトのフィールドを返すかどうかを示します。このフィールドには、VeriMove で必要とされる詳細結果インジケータ データが含まれます。このフィールドの内容については、VeriMove のユーザー ガイドを参照してください。次のいずれかです。</p> <p><b>Y</b>          フィールド VeriMoveDataBlock を返します。</p> <p><b>N</b>          フィールド VeriMoveDataBlock を返しません。</p>

**AdditionalInputData** について

ValidateAddress は、住所正規化プロセスにおいて、一部の入力データを無視します。この余分なデータ ("ドロップ データ" と呼ばれることもあります) は、AdditionalInputData に返されます。ドロップ データの例としては、次のものがあります。

- 配達指示 ("勝手口に置いてください" など)
- 電話番号 ("555-135-8792" など)
- 注意書き ("Attn: John Smith" など)

このようなデータは通常、住所に混在していることはありません。混在している場合、ValidateAddress はほとんどの場合にこの余分なデータを認識することができ、AdditionalInputData に返します。

**注:** ValidateAddress は、split indicia 住所からのドロップ データを返しません。split indicia 住所とは、主要な住所が複数の住所行に分割されている住所のことです。例えば、主要な住所が "1 Green River Valley Rd" である場合、次のようになります。

```
1 Green River
Valley Rd
01230
```

住所に複数のドロップ データがある場合、各データは、米国住所の場合はセミコロンと空白 ("; ")、米国外の住所の場合は空白で区切られます。AdditionalInputData におけるドロップ データの順序は、次のようになります。

1. care of (気付)、mail stop (郵便物集配所)(米国住所のみ)
2. 住所行に検出されたその他の余分なデータ
3. まったく未使用のデータ行

例えば、入力住所が次のとおりであるとします。

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

この場合、AdditionalInputData には次のデータが含まれます。

C/O John Smith; Apt 5 Drop At Back Dock; 555-123-4567; Jsmith@g1.Com; 555-123-4567

ValidateAddress では、以下の種類の余分なデータを処理できます。

#### Care Of (気付) データ

米国住所に対しては、"care of" データが AdditionalInputData に返されます。以下の住所には、"care of" データの例が含まれています。

123 Main St C/O John SmithApt 505674

123 Main StApt 5 ATTN John Smith05674

123 Main St Apt 5MailStop 205674

#### 独立した住所行に存在する余分なデータ

ValidateAddress は、米国およびカナダの住所に対し、独立した住所行に余分なデータを返します。

米国住所に対しては、住所行の空白でない最初の 2 行を使用して、住所の正規化を行います。ただし、企業名抽出または都市化コード抽出のオプションが有効である場合を除きます (詳細については、「[米国住所の住所行処理 \(572ページ\)](#)」を参照してください)。他の住所行に存在するデータは、AdditionalInputData に返されます。以下の住所において、"John Smith" は AdditionalInputData に返されます。"John Smith" は空白でない 3 つめの住所行に存在しており、ValidateAddress は米国住所に対して、空白でない最初の 2 つの住所行のみを使用するためです。

123 Main StApt 5John Smith05674

空白でない最初の 2 つの住所行に余分なデータが含まれる場合、そのデータは AdditionalInputData に返されます。例えば、以下の住所において、"John Smith" は AdditionalAddressData に返されます。

123 Main StJohn Smith05674

John Smith123 Main St05674

以下の住所では、"John Smith" と "Apt 5" の両方が AdditionalAddressData に返されます。"John Smith" が返されるのは、これが、最初の 2 つの住所行の 1 つに存在する余分なデータであるためです。"Apt 5" が返されるのは、米国住所データは、空白でない最初の 2 行に記載しなければならないためです。

John Smith123 Main StApt 505674

## 住所行に混在する余分なデータ

住所行に混在する余分なデータは、`AdditionalInputData` に返されます。例えば、以下の住所において、"John Smith" は `AdditionalInputData` に返されます。

123 Main St John Smith05674

123 Main St Apt 5 John Smith05674

123 Main St John SmithApt 505674

123 Main StApt 5 John Smith05674

米国住所に対しては、住所行の末尾に存在する余分なデータのみが `AdditionalInputData` に返されます。米国住所において、住所行の末尾以外に存在する余分なデータは返されません。例えば、以下の住所において、"John Smith" は返されません。

John Smith 123 Main St05674

123 Main John Smith St05674

マッチさせるために通り名が変更され、通り名または接尾語が行の末尾にあった場合は、`AdditionalInputData` には、元の通り名や接尾語が含まれることがあります。例えば、次の住所があるとします。

Pitney Bowes  
4200 ParliamentLanham MD

`ValidateAddress` は、ストリート名の綴りを修正し、接尾語を追加して、修正済みのストリートの住所として "4200 Parliament Pl" を返し、"Parliament" を `AdditionalInputData` に返します。

## 二重住所

二重住所とは、通り情報と、PO Box/地方配送路/幹線請負契約情報の両方を含む住所のことです。選択した処理オプションに応じて、住所の正規化に使用されない二重住所の一部が、`AdditionalInputData` に返される場合があります。詳細については、[二重住所ロジックについて](#) (588ページ) を参照してください。

## ValidateAddressGlobal

`ValidateAddressGlobal` は、米国およびカナダ以外の住所の標準化と検証のみを制御します。`ValidateAddressGlobal` は、米国およびカナダの住所の妥当性も確認できますが、その他の国の住所の妥当性を確認する能力に優れています。米国およびカナダ以外の住所を大量に処理する場合は、`ValidateAddressGlobal` の使用を検討してください。

`ValidateAddressGlobal` は `Universal Addressing` モジュールの一部です。

`ValidateAddressGlobal` は、書き直し、パーシング、バリデーション、書式設定など、いくつもの手順を実行して、住所の品質を高めています。

### 文字セットのマッピングと書き直し

`ValidateAddressGlobal` は他国の文字列と、それらの複雑な問題を処理します。Unicode に完全対応の文字列処理を使用するため、アルファベット以外の文字をラテン文字セットに書き直したり、異なる文字セット間でマッピングしたりできます。

文字セットのマッピング、および書き直しについて、以下の機能があります。

- UTF-8、ISO 8859-1、GBK、BIG5、JIS、EBCDIC など、30 以上の文字セットをサポート
- 言語のルールに従って、付加記号を正しく "除去"
- さまざまなアルファベットをラテン スクリプトに書き直し
- ギリシャ文字 (BGN/PCGN 1962、ISO 843 - 1997)
- キリル文字 (BGN/PCGN 1947、ISO 9 - 1995)
- ヘブライ語
- 日本語の片仮名、平仮名、漢字
- 中国語のピンイン (標準中国語、 広東語)
- 韓国語のハングル文字

### 住所のパーシング、書式設定、および正規化

住所データのフィールド入力の誤りを再構成することは、特に他国の住所で行う場合、複雑で難しい作業です。住所データをコンピュータのシステムに入力する際、曖昧になってしまう部分が多いからです。特に問題なのが、(企業や個人名を通りの住所フィールドに入力するなど) 要素を誤ったフィールドに入力したり、省略形を使用する場合に、言語固有だけでなく、国固有の省略形に変えてしまうケースです。`ValidateAddressGlobal` は住所行の住所要素を識別し、正しいフィールドに割り当てます。これは実際の検証前に行う重要な作業です。再構成を行わなければ、"一致が見つからない" という結果になる可能性があります。

住所要素の正しい識別は、特定のフィールド長要件に合わせて住所を切り捨てたり、短縮しなければならない場合にも重要です。正しい情報が正しいフィールドに割り当てられていれば、特定の切り捨てルールを適用することができます。

- 住所行をパースおよび解析し、個々の住所要素を識別
- 30 を越える文字セットを処理
- 宛先国の郵便ルールに従って住所の書式を整える
- 住所要素を正規化 (AVENUE を AVE に変更するなど)

### Global Address 検証

住所の検証は、正しくパースされた住所データを郵便組織または他のデータ プロバイダが提供する参照データベースと比較する訂正処理です。`ValidateAddressGlobal` は、洗練されたファジー マッチング テクノロジーを使用して個々の住所要素を検証し、正しいことを確認するとともに、郵便規格とユーザの優先設定に基づいて出力を正規化および書式設定します。FastCompletion 検



証タイプは、簡易住所入力アプリケーションに使用できます。いくつかの住所フィールドには切り捨てられたデータを入力することができ、この入力に基づいて提案を生成します。

住所を完全に検証できない場合もあります。**ValidateAddressGlobal** には、配達可能性によって住所を分類する、ユニークな配達可能性評価機能があります。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

```
{
  "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": []
  }}}

```

### XML 応答の例

次の例では、XML 応答を要求します。

```

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

```

この要求から次のような XML が返されます。

```

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

```

```

<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>

```

### 要求

#### 入力データのパラメータ

`ValidateAddressGlobal` は、入力として標準住所を受け取ります。どの国の住所であるかにかかわらず、すべての住所がこのフォーマットを使用します。

表 50 : `ValidateAddressGlobal` の入力

パラメータ	書式	説明
Data.AddressLine1 から Data.AddressLine6	文字列 [79]	これらのフィールドには住所行データが格納されます。AddressLine1には最初の住所行、AddressLine2には2行目の住所行が格納されます。以降もこれと同様になります。都市、州/省、および郵便番号情報は、住所行フィールドではなく、それぞれのフィールドに配置する必要があります。例:  <b>AddressLine1:</b> 17413 Blodgett Road <b>AddressLine2:</b> PO Box 123 <b>City:</b> Mount Vernon <b>StateProvince:</b> WA <b>PostalCode:</b> 97273 <b>Country:</b> USA  入力住所が適切な住所行および City、StateProvince、PostalCode フィールドにまだパースされていない場合は、住所行フィールドの代わりに UnformattedLine フィールドを使用してください。
Data.City	文字列 [79]	都市名
Data.StateProvince	文字列 [79]	州または省。

パラメータ	書式	説明
Data.PostalCode	文字列 [79]: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999	住所の郵便番号。米国では、ZIP Code®になります。
Data.Contact	文字列 [79]	受取人の名前。例えば、"Mr.Jones"。
Data.Country	文字列 [79]	国名。または オプションに値が指定されていない場合、国を指定する必要があります。
Data.FirmName	文字列 [79]	会社名または企業名。
Data.Street	文字列 [79]	Street
Data.Number	建物 [79]	番号
Data.Building	文字列 [79]	建物
Data.SubBuilding	文字列 [79]	SubBuilding
Data.DeliveryService	文字列 [79]	配送サービス

パラメータ	書式	説明
Data.UnformattedLine1 から Data.UnformattedLine10	文字列 [79]	<p>入力住所が完全にパースされていない場合、かつ <code>ValidateAddressGlobal</code> によって住所を適切なフィールドにパースしたい場合は、このフィールドを使用してください。例:</p> <p><b>UnformattedLine1:</b> 17413 Blodgett Road<b>UnformattedLine2:</b> PO Box 123<b>UnformattedLine3:</b> Mount Vernon WA 97273<b>UnformattedLine4:</b> USA</p> <p>この住所は、以下の出力フィールドにパーシングされます。</p> <p><b>AddressLine1:</b> 17413 Blodgett Road<b>AddressLine2:</b> PO Box 123<b>City:</b> Mount Vernon<b>StateProvince:</b> WA<b>PostalCode:</b> 97273<b>Country:</b> USA</p> <p>注：フォーマットされていない行フィールドへの入力を指定した場合は、住所全体をフォーマットされていない行フィールドだけを使用して指定する必要があります。City や StateProvince など、その他のフィールドを、フォーマットされていない行フィールドと合わせて使用することはできません。</p>

## オプションのパラメータ 入力オプション

表 51 : `ValidateAddressGlobal` の入力オプション

パラメータ	説明/有効値
Option.Database.AddressGlobal	住所検証で使用する郵便データを含むデータベースリソースを指定します。指定できるのは、Management Console の <b>[グローバル データベース リソース]</b> パネルで定義されたデータベースに限られます。詳細については、『 <i>Spectrum™ Technology Platform 管理ガイド</i> 』を参照してください。
Option.Input.DefaultCountryISO3	入力レコードに明示的な国情報が含まれない場合に使用するデフォルトの国を指定します。指定する際、ISO3 の国コードを使用してください。デフォルトの国を指定しない場合、各入力レコードの Country 入力フィールドに国を指定する必要があります。ISO コードの一覧は、「 <a href="#">ISO 国コードとモジュール サポート (735ページ)</a> 」を参照してください。

パラメータ	説明/有効値
Option.Input.ForceCountryISO3	住所レコードが、常にここで指定された国から発送されたものとして扱われるようになります。住所レコードの国やデフォルトの国は上書きされます。指定する際、ISO3 の国コードを使用してください。ISO コードの一覧は、 <a href="#">ISO 国コードとモジュールサポート (735ページ)</a> を参照してください。
Option.Input.FormatDelimiter	<p>入力ファイルで複数行の住所に標準以外の書式設定を使用できます。このフィールドで指定できる値を次に示します。</p> <ul style="list-style-type: none"> <li>• CRLF (デフォルト)</li> <li>• LF</li> <li>• CR</li> <li>• SEMICOLON ( 2101 MASSACHUSETTS AVE NW ; WASHINGTON DC 20008 )</li> <li>• COMMA ( 2101 MASSACHUSETTS AVE NW , WASHINGTON DC 20008 )</li> <li>• TAB ( 2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008 )</li> <li>• PIPE ( 2101 MASSACHUSETTS AVE NW   WASHINGTON DC 20008 )</li> <li>• SPACE ( 2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008 )</li> </ul> <p>注：入力オプションと出力オプションの両方に同じ値を選択する必要があります。</p>

## 出力オプション

表 52 : **ValidateAddressGlobal** の出力オプション

パラメータ	説明
Option.Result.MaximumResults	このオプションには、返される候補住所の最大数を指定します。このフィールドは、バッチ処理には無効です。それ以外の処理モードの場合、デフォルト値は1、最大値は99です。FastCompletion モードを使用している場合は、1 よりも大きい数を入力して、フィールド入力のための選択肢が複数返されるようにするとよいでしょう。
Option.Result.IncludeInputs	<p>出力の中に、入力データを含めるかどうかを指定します。有効にすると、.Input で終わるフィールド (対応する入力フィールドを含む) が出力に含まれます。例えば、出力フィールドの AddressLine1.Input に、入力フィールドの AddressLine1 に指定したデータが含まれます。</p> <p><b>TRUE</b>                      入力データを出力に含めます。</p> <p><b>FALSE</b>                     出力に元の入力データを含めません (デフォルト)。</p>

パラメータ	説明
Option.Result.StateProvinceType	StateProvince フィールドのフォーマットを指定します。次のいずれかを選択します。 <b>ABBREVIATION</b> 州または省の省略形を返します。例えば、North Carolina であれば "NC" が返ります。 <b>COUNTRY_STANDARD</b> その国の郵便当局で使用されるフォーマットに応じて、省略形または完全な名前が返ります（デフォルト）。 <b>EXTENDED</b> 州または省の (省略形ではなく) 完全な名前が返ります (例えば、"North Carolina" など)。

パラメータ	説明
Option.Result.CountryType	ValidateAddressGlobal から返される国名で使用する言語またはコードを指定します。
<b>ISO2</b>	その国の 2 文字の ISO コード
<b>ISO3</b>	その国の 3 文字の ISO コード
<b>ISO_NUMBER</b>	ISO 国番号
<b>NAME_CN</b>	中国語
<b>NAME_DA</b>	デンマーク語
<b>NAME_DE</b>	ドイツ語
<b>NAME_EN</b>	英語 (デフォルト)
<b>NAME_ES</b>	スペイン語
<b>NAME_FI</b>	フィンランド語
<b>NAME_FR</b>	フランス語
<b>NAME_GR</b>	ギリシャ文字
<b>NAME_HU</b>	ハンガリー語
<b>NAME_IT</b>	イタリア語
<b>NAME_JP</b>	日本語
<b>NAME_KR</b>	韓国語
<b>NAME_NL</b>	オランダ語
<b>NAME_PL</b>	ポーランド語
<b>NAME_PT</b>	ポルトガル語
<b>NAME_RU</b>	ロシア語
<b>NAME_SA</b>	サンスクリット語
<b>NAME_SE</b>	スウェーデン語

## パラメータ

## 説明

**Option.Result.PreferredScript** 出力がどのアルファベットで返されるかを指定します。データがどのアルファベットで返されるかは、国によって異なります。言語の設定で何を選択したかに関わらず、ほとんどの国で出力は **Latin I** になります。

**ASCII\_Extended** 特殊文字 (Ã– = OE など) で拡張された ASCII 文字

**ASCII\_Simplified** ASCII 文字

**Database** (デフォルト) Latin I または ASCII 文字 (参照データベースの標準による)

**Latin** Latin I 文字

**Latin\_Alt** Latin I 文字 (代替の書き直し)

**Postal\_Admin\_Alt** Latin I または ASCII 文字 (現地郵便局の代替)

**Postal\_Admin\_Pref** Latin I または ASCII 文字 (現地郵便局の選択による)

Latin I 以外のアルファベットを使用する国の場合、返されるアルファベットは国によって異なります。詳細については、[Latin I 以外のアルファベットを使用する国](#) (353ページ) を参照してください。

**Option.Result.PreferredLanguage** 出力がどの言語で返されるかを指定します。データがどのアルファベットで返されるかは、国によって異なりますが、ほとんどの国で、言語の設定に何を選択したかに関わらず、出力は **Latin I** になります。

**DATABASE** 各住所の参照データから得られた言語で返されます。こちらがデフォルトです。

**ENGLISH** 地方および州/省の名前が (可能であれば) 英語で出力されません。

**Option.Result.Casing**

出力の大文字と小文字の区別を指定します。

**NATIVE** 参照データベースの標準に基づいて出力されます。

**UPPER** すべての国で、大文字で出力されます。

**LOWER** すべての国で、小文字で出力されます。

**MIXED** 国固有のルールに従って大文字と小文字が判断されます。

**NOCHANGE** パースモードの場合、データは入力された方法で返されます。バリデーションモードの場合、大文字と小文字の区別には参照データに見つかったものや、郵便のルールに従ったものを使用しません。参照データを使用してチェックできなかった値は、入力時の大文字と小文字の区別を保持します。



## パラメータ 説明

**Option.Result.FormatDelimiter** 出力で複数行の住所に標準以外の書式設定を使用できます。このフィールドで指定できる値を次に示します。

- CRLF (デフォルト)
- 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)

注：入力オプションと出力オプションの両方に同じ値を選択する必要があります。

### Latin 1 以外のアルファベットを使用する国

Latin I 以外のアルファベットを使用する国の場合、返されるアルファベットは国によって異なります。それらの国々で、出力がどのように返されるかを以下の表に示します。ここに含まれないすべての国では、フィールド オプションに指定された値が使用されます。

Country	データ	ベース	ラテン文字
RUS	キリル文字	キリル文字	キリル文字
			CYRILLIC_ISO CYRILLIC_BGN CYRILLIC_ISO + LATIN CYRILLIC_ISO + LATIN_SIMPLE
JPN	漢字	漢字	かな
			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

Country データ  
ベース

ラテン文字

TWN	Hanzi	Hanzi	Hanzi	CHINESE_ CANTONESE	CHINESE_ MANDARIN	CHINESE_ CANTONESE + LATIN_SIMPLE	CHINESE_ CANTONESE + LATIN
GRC	ギリシャ文 字	ギリシャ文 字	ギリシャ文 字	GREEK_ISO	GREEK_BGN	GREEK_ISO + LATIN_SIMPLE	GREEK_ISO + LATIN
KOR	ラテン文字	ハングル	Hanja	KOREAN	KOREAN	KOREAN + LATIN_SIMPLE	KOREAN + LATIN
ISR	ラテン文字	ヘブライ語	ヘブライ語	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

プロセス オプション

表 53 : ValidateAddressGlobal のプロセス オプション

パラメータ	説明
Option.Process.OptimizationLevel	<p>このオプションを使用して、処理速度と品質が適切なバランスを保つよう設定します。次のいずれかです。</p> <p><b>NARROW</b> パーサーは、House Number を Street の情報から分割することを除き、入力の割り当てに厳密に従います。</p> <p><b>STANDARD</b> パーサーは、以下に示すように住所要素をよりアクティブに分割します。</p> <ul style="list-style-type: none"> <li>• Province は Locality の情報から分割される</li> <li>• PostalCode は Locality の情報から分割される</li> <li>• House Number は Street の情報から分割される</li> <li>• SubBuilding は Street の情報から分割される</li> <li>• DeliveryService は Street の情報から分割される</li> <li>• SubBuilding は Building の情報から分割される</li> <li>• Locality は PostalCode の情報から分割される</li> </ul> <p><b>WIDE</b> パーサーによる分割は Standard に似た方法で行われますが、追加で最大 10 のパーシング候補が検証に渡されて処理されます。検証は検索のツリーを広げ、参照データのエントリを対象に追加してマッチングを行います。</p> <p>最適化レベルを調整しても、ここで説明したような分割で必要とされる郵便参照データ情報が揃っていない国においては、効果が得られない場合があります。</p> <p>分類の精度を Narrow から Standard に増やしても処理パワーはある程度消費されますが、最適化レベルを Wide に設定し、検索ツリーを大きくして検証を行った場合、指定された入力データから最も精密な結果を得ようとデータのアクセスや比較の件数が増加するため、処理速度が非常に遅くなります。</p>

## パラメータ 説明

---

Option.Process.Mode	住所に対して実行する処理のタイプを指定します。次のいずれかです。
<b>BATCH</b>	<p>このモードは手動でのデータの入力または選択が不可能なバッチ処理環境で使用します。処理速度を重視して最適化されているため、自動で修正できないあいまいなデータが見つかった場合は、住所修正の試行が停止します。指定の国がデータベースに見つからない場合、バッチ処理モードはパースモードに変更されます。</p> <p>注：プロセスステータスとして I3 の値が返された場合、試行は失敗とみなされ、ステータスとして F の値が返されます。</p>
<b>CERTIFIED</b>	<p>このモードはオーストラリア郵便当局のバッチ処理環境で使用します。Validate Address Global はオーストラリア郵便公社の Address Matching Approval System (AMAS) で認定されています。これは郵便住所ファイルに照らして郵便の正規化と検証を行うことで、郵便料金の割引と不達郵便物の最小化を実現します。</p>
<b>FASTCOMPLETION</b>	<p>高速実行モードは、切り捨て処理されたデータを住所フィールドに入力して、Validate Address Global による提案の生成を行いたい場合に使用します。例えば、コールセンターや店頭などで作業をしている場合に住所要素の一部だけを入力し、高速実行の機能を使用して、完全な住所の候補を正しく入手することができます。</p>
<b>INTERACTIVE</b>	<p>このモードは、インタラクティブな環境で処理を行い、住所入力があいまいな場合に提案を生成する場合に使用します。このタイプの検証は、とりわけ顧客や潜在顧客から入手したデータを入力した環境で使用されます。住所がほぼ完全に入力されている必要があり、その入力データの検証や修正を試行します。あいまいなデータが検出された場合、この検証タイプでは最大で 20 の提案が生成され、それを選択候補の一覧として使用できます。指定の国がデータベースに見つからない場合、インタラクティブモードはパースモードに変更されます。</p>
<b>PARSE</b>	<p>このモードは住所入力をトークンに分割し、そのあとに他のシステムで行われる処理に送る場合、すなわち検証をバイパスする場合に使用します。例えば、住所データの品質がすでに高く、単に素早くトークンに分割して外部のシステムにエクスポートするか、あるいは下流のステージで使用する場合にこのモードを利用できます。</p>

---

## パラメータ 説明

Option.Process.MatchingScope	住所が検証されるために、住所がどの程度参照データと一致する必要があるかを指定します。次のいずれかです。  注：以下の設定は、必要とされる詳細情報が郵便参照データに見つからない国に対しては効果がない場合があります。
<b>ALL</b>	すべての住所要素が一致する必要があります。
<b>DELIVERYPOINT_LEVEL</b>	Validate Global Address で州/省、郵便番号、都市/地方/郊外、通り、家番号、および従属する建物が一致する必要があります。
<b>STREET_LEVEL</b>	Validate Global Address で州/省、郵便番号、都市/地方/郊外、および通りが一致する必要があります。
<b>LOCALITY_LEVEL</b>	Validate Global Address で州/省、郵便番号、および都市/地方/郊外が一致する必要があります。

## 応答

### 住所データ

表 54 : パース済み住所要素

## 応答要素 説明

AddressBlock1-9	<p>AddressBlock 出力フィールドには、正規化済み、または標準化済みの住所が、実際の郵便物に印刷される形式にフォーマットされて入ります。Validate Address Global は住所を郵便当局の規格に従って住所ブロックにフォーマットします。住所の各行が、別々の住所ブロックフィールドに入れて返されます。AddressBlock1 から AddressBlock9 まで、最大 9 つの住所ブロック出力フィールドが使用されません。例えば、以下の入力住所の場合、</p> <p>AddressLine1: 4200 Parliament PlaceAddressLine2: Suite 600City: LanhamStateProvince: MDPostalCode: 20706</p> <p>以下の住所ブロックが出力されます。</p> <p>AddressBlock1: 4200 PARLIAMENT PL STE 600AddressBlock2: LANHAM MD 20706-1882</p>
-----------------	--

応答要素	説明
AddressLine1-6	<p>住所が検証された場合、住所行フィールドには検証済みで正規化済みの住所行が入ります。住所が検証できなかった場合、住所行フィールドには入力された住所が変更されずに入ります。ただし、住所の最後の行は LastLine フィールドに入ります。例:</p> <p>AddressLine1: 4200 PARLIAMENT PL STE 600LastLine: LANHAM MD 20706-1882</p>
AdministrativeDistrict	州/省より小さいが、都市よりも大きいエリア
ApartmentLabel	フラットまたはユニット タイプ (STE や APT など)。例: 123 E Main St <b>Apt 3</b>
ApartmentNumber	フラットまたはユニット番号。例: 123 E Main St <b>Apt 3</b>
BlockName	地所やブロックの名前。
BuildingName	建物の名前。例えば <b>Sears Tower</b> など。
City	都市の名前。例えば <b>Vancouver, BC</b> など。
City.AddInfo	都市の追加情報。
City.SortingCode	例えばプラハやダブリンなど、特定の国の広い地方で、郵便当局が配達時間を短縮するために使用するコード。
Contact	受取人の名前。例えば、 <b>Mr. Jones</b> など。
Country	国が、言語または オプションで指定されたコードで入ります。
County	州や省に従属する情報。州や省をさらに細かく分割します。米国の郡はその一例です。
FirmName	会社名。
Floor	部屋番号やアパート番号など、建物をさらに細かく分割する情報。例: 123 E Main St Apt 3, <b>4th Floor</b>

応答要素	説明
HouseNumber	家番号 1。例: 298A-1B New South Head Rd
LastLine	完成された最終の住所行 (都市、州/省、および郵便番号)。
LeadingDirectional	通り名の前に付けて通りの方向を表します。例えば、138 N Main Street の N がこれに該当します。
Locality	場所の名前に従属し、地方をさらに細かく分割します。例としては、メキシコの Colonia や、スペインの Urbanisaciones などがあります。
POBox	郵便受けの記述子 (POBox、Postfach、Case Postale など) と番号。
PostalCode	住所の郵便番号。郵便番号のフォーマットは国によって異なります。
PostalCode.AddOn	郵便番号の 2 番目の部分。例えば、カナダの住所ではこれは LDU です。米国住所に対しては、これは ZIP+4 アドオンです。ほとんどの国ではこのフィールドを使用しません。
PostalCode.Base	郵便番号の基本部分。
Room	建物の部屋番号。
SecondaryStreet	補助的な通りまたは地方集配路の名前。
StateProvince	州または省の名前。
StreetName	建物が存在する通りの名前。例: 123 E Main St Apt 3
StreetSuffix	通り接尾語。例: 123 E Main St Apt 3
SubBuilding	部屋番号など建物の一部。例えば Suite 102 など。
Suburb	場所の名前に従属し、地方をさらに細かく分割します。例えばトルコの Mahalle など。

応答要素	説明
Territory	地域の名前。地域は州/省より大きいものです。
TrailingDirectional	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>

### 元の入力データ

このオプションは、元の入力データを <フィールド名>.Input フィールドに出力します。

**表 55 : 元の入力データ**

応答要素	書式	説明
AddressLine1.Input	文字列 [79]	最初の住所行。
AddressLine2.Input	文字列 [79]	2 行目の住所行。
AddressLine3.Input	文字列 [79]	3 行目の住所行。
AddressLine4.Input	文字列 [79]	4 行目の住所行。
AddressLine5.Input	文字列 [79]	5 行目の住所行。
AddressLine6.Input	文字列 [79]	6 行目の住所行。
City.Input	文字列 [79]	都市名



応答要素	書式	説明
StateProvince.Input	文字列 [79]	州または省
PostalCode.Input	文字列 [79]:	住所の郵便番号。米国では、ZIP Code になります。次のいずれかのフォーマットです。 99999 99999-9999 A9A9A9 A9A 9A9 9999 999
Contact.Input	文字列 [79]	受取人の名前。例えば、"Mr.Jones"。
Country.Input	文字列 [79]	入力为国フォーマットに選択したフォーマットを使用して国を指定します (英語名、ISO コード、または UPU コード)。有効な値の一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。
FirmName.Input	文字列 [79]	会社名または企業名。
Street.Input	文字列 [79]	Street
Number.Input	建物 [79]	番号
Building.Input	文字列 [79]	建物
SubBuilding.Input	文字列 [79]	SubBuilding

応答要素	書式	説明
DeliveryService.Input	文字列 [79]	配送サービス

## 結果コード

これらの出力フィールドには、検証処理の結果に関する情報が格納されます。

表 56 : 結果コード

応答要素	結果コード
AddressType	<p>米国とカナダの住所の場合のみ、AddressType フィールドは住所のタイプを示します。次のいずれかです。</p> <p><b>F</b> 住所の妥当性が確認され社名まで修正されました。</p> <p><b>B</b> 住所の妥当性が確認され建物名まで修正されました。</p> <p><b>G</b> 住所は局留めの住所です。</p> <p><b>H</b> 住所の妥当性が確認され高層のデフォルトまで修正されました。</p> <p><b>L</b> 住所は、Large Volume Receiver (LVR) です。</p> <p><b>M</b> 住所は、軍施設の住所です。</p> <p><b>P</b> 住所の妥当性が確認され私書箱まで修正されました。</p> <p><b>R</b> 住所の妥当性が確認され地方配送路まで修正されました。</p> <p><b>S</b> 住所の妥当性が確認され通りの住所まで修正されました。</p> <p><b>U</b> タイプが不明なために住所を検証/修正できませんでした。</p>
Confidence	<p>返された住所に割り当てられた確信レベル。範囲は 0 ~ 100 です。0 は失敗を表し、100 はマッチ結果が正しいことに対する確信レベルが非常に高いことを表します。</p>
CountOverflow	<p>候補住所数が返される数より多いかどうかを示します。次のいずれかです。</p> <p><b>はい</b> 他の候補住所があります。他の候補を取得するには、 の値を増やします。</p> <p><b>いいえ</b> いいえ、これ以外の候補はありません。</p>

## 応答要素

## 結果コード

---

**ElementInputStatus**

**ElementInputStatus** は、入力要素を参照データに対してマッチングした際の要素ごとの情報を提供します。このフィールドの値は、バッチとパースのどちらのモードを使用するかによって変わります。このフィールドの値については、[ElementInputStatus](#)、[ElementResultStatus](#)、および [ElementRelevance](#) の解釈 (367ページ) を参照してください。

---

**ElementRelevance**

その土地の郵便当局の基準から基準から判断して、実際にどの住所要素が関連しているかを示します。このフィールドの値については、[ElementInputStatus](#)、[ElementResultStatus](#)、および [ElementRelevance](#) の解釈 (367ページ) を参照してください。

---

**ElementResultStatus**

**ElementResultStatus** は、出力フィールドが入力フィールドから変更されているか、またどのように変わっているかを示すことによって、結果を **ProcessStatus** フィールドよりも詳細に分類します。このフィールドの値については、[ElementInputStatus](#)、[ElementResultStatus](#)、および [ElementRelevance](#) の解釈 (367ページ) を参照してください。

---

**MailabilityScore**

郵便が住所にどの程度確実に配達されるかの予測を示します。次のいずれかです。

- 5 確実に配達される
- 4 ほぼ確実に配達される
- 3 おそらく配達される
- 2 五分五分
- 1 配達されない可能性が高い
- 0 配達されない

---

**ModeUsed**

使用された処理モードを示します。処理モードは、オプションで指定します。モードの説明については、[プロセス オプション](#) (667ページ) を参照してください。

---

**MultimatchCount**

住所が参照データ内の複数の候補住所と一致した場合、このフィールドには一致した候補の数が含まれます。

---

応答要素

結果コード

---

ProcessStatus

## 応答要素

## 結果コード

出力の品質に関する一般的な記述が含まれます。出力の品質について、詳しくは `ElementResultStatus` フィールドを参照してください。

次のいずれかです。

- V4** 検証済み。正しい入力データです。すべての要素がチェックされ、入力データが完全に一致しました。
- V3** 検証済み。入力されたデータは正しいものですが、一部またはすべての要素が正規化されたか、あるいは入力データに旧式の名前または外名 (エクソニム) が含まれます。
- V2** 検証済み。正しい入力データですが、一部の要素について、参照データが完全でないために検証できていません。
- V1** 検証済み。正しい入力データですが、ユーザによる正規化によって、正しく配達される可能性が損なわれています (たとえば選択された郵便番号の長さが短すぎるなど、ユーザによって要素の正規化が正しく行われていない)。検証によってセットされません。
- C4** 修正済み。すべての要素がチェック済みです。
- C3** 修正済みですが、一部の要素をチェックできませんでした。
- C2** 修正済みですが、配達のスレータスがわかりません (参照データがありません)。
- C1** 修正済みですが、ユーザによる正規化が正しくなかったため、配達のスレータスがわかりません。検証によってセットされません。
- I4** データを完全には修正できませんでした。かなりの確率で正しく配達されます。単一の一致が見つかりました (例えば HNO は誤りだが、参照データに HNO が 1 件だけ見つかるような場合)。
- I3** データを完全には修正できませんでした。かなりの確率で正しく配達されます。複数の一致が見つかりました (例えば HNO は誤りだが、参照データに複数の HNO が見つかるような場合)。
- I2** データを修正できませんでした。この住所で正しく配達される可能性がわずかにあります。
- I1** データを修正することができず、正しく配達される可能性もほとんどありません。
- RA** 国が強制国の設定から認識されました。
- R9** 国が `DefaultCountryISO3` の設定から認識されました。
- R8** 国が名前からエラーなしで認識されました。
- R7** 国が名前からエラー付きで認識されました。
- R6**

## 応答要素

## 結果コード

---

	国が地域から認識されました。
<b>R5</b>	国が省から認識されました。
<b>R4</b>	国が主要な都市から認識されました。
<b>R3</b>	国がフォーマットから認識されました。
<b>R2</b>	国がスクリプトから認識されました。
<b>R1</b>	国が認識されませんでした。複数の一致が見つかりました。
<b>R0</b>	国が認識されませんでした。
<b>S4</b>	完全にパースされました。
<b>S3</b>	パースされ、複数の結果が得られました。
<b>S2</b>	パースされ、エラーが発生しました。要素の位置が変わっています。
<b>S1</b>	パース エラー。入力フォーマットが一致しません。
<b>N1</b>	バリデーションエラー: 国が認識されなかったため、検証が実行されませんでした。
<b>N2</b>	バリデーションエラー: 必要な参照データベースを使用できないため、検証が実行されませんでした。
<b>N3</b>	バリデーションエラー: 国をロック解除できなかったため、検証が実行されませんでした。
<b>N4</b>	バリデーションエラー: 参照データベースが破損しているか、フォーマットが正しくないため、検証が実行されませんでした。
<b>N5</b>	バリデーションエラー: 参照データベースが古すぎるため、検証が実行されませんでした。
<b>N6</b>	バリデーションエラー: 入力データが十分でないため、検証が実行されませんでした。
<b>Q3</b>	高速実行ステータス: 提案 (完全な住所) を利用可能です。
<b>Q2</b>	高速実行ステータス: 完全な住所が提案されましたが、入力データの要素と組み合わせられています (追加または削除)。
<b>Q1</b>	高速実行ステータス: 提案された住所は完全ではありません (情報を追加してください)。
<b>Q0</b>	高速実行ステータス: 十分でない情報から提案が生成されました。

---

## 応答要素

## 結果コード

**Status**

処理試行が成功したか失敗したかをレポートします。

**NULL**

成功

**F**

失敗

**Status.Code**

失敗したものがあれば、その理由を示します。

**Status.Description**

失敗したものがあれば、その理由を説明する記述が入ります。

**ElementInputStatus**、**ElementResultStatus**、および **ElementRelevance** の解釈

**ElementInputStatus**、**ElementResultStatus**、および **ElementRelevance** の出力フィールドには、検証操作の結果を詳細に示す一連の数字が含まれます。**ElementInputStatus** にはパーシング操作に関するいくつかの情報が含まれます。

**ElementInputStatus** の値は以下のようになります。

44606040600000000060

**ElementResultStatus** の値は以下のようになります。

88F0F870F00000000040

**ElementRelevance** の値は以下のようになります。

11101010100000000000

これらのフィールドの値について理解するためには、各ポジションの数字がどの要素を表すか、および各ポジションの値の意味を知る必要があります。たとえば、先頭の数字は **PostalCode.Base** 出力フィールドの結果を示します。以下に各ポジションの意味を一覧で示します。

- ポジション 1—**PostalCode.Base**
- ポジション 2—**PostalCode.AddOn**
- ポジション 3—**City**
- ポジション 4—**Locality** および **Suburb**
- ポジション 5—**StateProvince**
- ポジション 6—**County**
- ポジション 7—**StreetName**
- ポジション 8—**SecondaryStreet**
- ポジション 9—**HouseNumber**
- ポジション 10—**Number** レベル 1

- ポジション 11—POBox
- ポジション 12—Delivery サービス レベル 1
- ポジション 13—Building レベル 0
- ポジション 14—BuildingName
- ポジション 15—Sub building レベル 0
- ポジション 16—Floor and Room
- ポジション 17—FirmName
- ポジション 18—Organization レベル 1
- ポジション 19—Country
- ポジション 20—Territory

ElementInputStatus の場合、検証の値として以下のいずれかが入ります。

- 0—空
- 1—見つからない
- 2—チェックなし (参照データなし)
- 3—誤り - 検証によってのみセット。参照データベースによれば、Number または DeliveryService のいずれかの数が正しい範囲内でない。入力はコピーされ、バッチモードでは修正されていない。インタラクティブ モードと FastCompletion では、提案が提供されている
- 4—この要素内でエラーありで一致
- 5—変更ありで一致 (挿入または削除)。例:
  - パーシング: "MainSt 1" の家番号を分割するなど
  - バリデーション: 外名 (エクソニム) が入力された場合に置換したり、フィールドのサイズを超えて入力され、国の参照データベースによって無効とされたデータを短くしたなど
- 6—エラーなしで一致

ElementInputStatus の場合、パーシングの値には以下のいずれかが入ります。

- 0—空
- 1—要素の位置を変更する必要があった
- 2—一致したが、正規化する必要があった
- 3—一致した

ElementRelevance の場合、パーシングの値には以下のいずれかが入ります。

- 0—空
- 1—要素の位置を変更する必要があった
- 2—一致したが、正規化する必要があった
- 3—一致した



ElementResultStatus の場合、(国を除くすべての住所要素に対して)以下のいずれかの値が入りません。

- 0—空
- 1—検証されず、変更されていない。元のデータがコピーされている
- 2—検証されなかったが正規化された
- 3—検証されたが、入力が無効(データベースによれば、数値が正しい範囲内でない)であるため変更されなかった入力がコピーされ、修正されていない(このステータス値はバッチモードでのみセットされる)
- 4—検証されたが、参照データが見つからないため変更されていない
- 5—検証されたが、複数の一致が見つかったため変更されていない。バッチモードでのみセットされる。そうでない場合、入力を置換する複数の提案が修正済みとしてマークされる(ステータス値 7)
- 6—検証され、入力値の除外という変更が行われている
- 7—検証され、参照データに基づく修正という変更が行われている
- 8—検証され、参照データに基づく値の追加という変更が行われている
- 9—検証され、変更はされていないが、配達ステータスがわからない(例えば、DPV の値で、指定の数値範囲が参照データと部分的にしか一致せず正しくないなど)
- C—検証され、妥当性確認済みだが名前が旧式であるため変更されている
- D—検証され、妥当性確認済みだが外名から公式の名前に変更されている
- E—検証され、妥当性確認済みだが大文字と小文字の区別、または言語に基づく正規化のため変更されている。入力が別の言語と完全に一致する場合に、検証でのみこのステータスがセットされる
- F—検証され、妥当性確認済みで、完全に一致したため変更されなかった

国(ポジション 19 と 20)に関しては、以下のいずれかの値が入ります。

- 0—空
- 1—国が認識されない
- 4—国が DefaultCountryISO3 の設定から認識された
- 5—国が認識されない。複数の一致が見つかった
- 6—国がスクリプトから認識された
- 7—国がフォーマットから認識された
- 8—国が主要な都市から認識された
- 9—国が省から認識された
- C—国が地域から認識された
- D—国がエラーありで名前から認識された
- E—国が名前からエラーなしで認識された
- F—国が ForceCountryISO3 の設定から認識された

## ValidateAddressLoqate

ValidateAddressLoqate は、郵便当局の住所データを使用して、住所を正規化し、妥当性を確認します。ValidateAddress Loqate は、情報を修正し、管轄の郵便当局が推奨する書式で住所の書式を整えることができます。また、郵便番号、都市名、州/省名など、欠落している郵便情報を追加します。

ValidateAddressLoqate は、ValidateAddressLoqate が住所の妥当性を確認したかどうか、返された住所の確信レベル、住所の妥当性が確認できなかった場合はその理由など、検証処理に関する結果インジケータも返します。

ValidateAddressLoqate は、住所のマッチングと正規化において、住所行をコンポーネントに分割し、それらを Universal Addressing モジュールの各種データベースの内容と比較します。マッチを検出した場合、入力住所をデータベース情報に合わせて正規化します。データベースにマッチしなかった場合、ValidateAddressLoqate は、オプションで入力住所の書式を整えます。書式設定プロセスでは、該当する郵便当局の規則に従って住所行の構成を試みます。

ValidateAddressLoqate は、Universal Addressing モジュールに含まれています。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

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

この要求から次のような JSON が返されます。

```
{"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": []
  }}}

```

### XML 応答の例

次の例では、XML 応答を要求します。

```

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

```

この要求から次のような XML が返されます。

```

<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>

```

## 要求

## 入力データのパラメータ

表 57: 入力フォーマット

パラメータ	書式	説明
Data.AddressLine1	文字列	最初の住所行。
Data.AddressLine2	文字列	2 行目の住所行。
Data.AddressLine3	文字列	3 行目の住所行。
Data.AddressLine4	文字列	4 行目の住所行。
Data.City	文字列	都市名。
Data.Country	文字列	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2 文字の ISO 3166-1 Alpha 2 国コード</li> <li>• 3 文字の ISO 3166-1 Alpha 3 国コード</li> <li>• 英語の国名</li> </ul> <p>ISO コードの一覧については、<a href="#">ISO 国コードとモジュール サポート (735ページ)</a> を参照してください。</p>
Data.FirmName	文字列	会社名または企業名。
Data.PostalCode	文字列	<p>住所の郵便番号は、次のフォーマットのいずれかで表されます。</p> <p>99999 99999-9999 A9A9A9 A9A 9A9 9999 999</p>

パラメータ	書式	説明
Data.StateProvince	文字列	州または省。

## 米国住所の住所行処理

米国住所の場合、AddressLine1 から AddressLine4 の入力フィールドの処理方法は、企業名抽出または都市化コード抽出のオプションが有効になっているかどうかによって異なります。2つのオプションのいずれかが有効になっている場合、ValidateAddressLoqate は、4 つすべてのフィールドのデータを参照して、住所の妥当性を確認し、要求されたデータ (企業名または都市化コード) を抽出します。どちらのオプションも有効でない場合は、ValidateAddressLoqate は、空白でない最初の 2 つの住所行フィールドのみを使用して、妥当性を確認します。他の住所行フィールドのデータは、AdditionalInputData 出力フィールドに返されます。例を次に示します。

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

この住所において、企業名抽出または都市化コード抽出が有効である場合、ValidateAddressLoqate は、4 つすべての住所行を確認します。企業名抽出と都市化コード抽出がどちらも有効でない場合は、ValidateAddressLoqate は、AddressLine1 と AddressLine3 (空白でない最初の 2 つの住所行) を参照して、そのデータを使用して住所の妥当性を確認します。AddressLine4 のデータは、AdditionalInputData 出力フィールドに返されます。

### オプション

以下の表に、ValidateAddressLoqate が返す情報の種類を制御するオプションの一覧を示します。

**表 58 : 出力データ オプション**

パラメータ	説明
Option.Database.Loqate	国際住所の妥当性の確認に使用するデータベースを指定します。国際住所検証用のデータベースを指定するには、【データベース】ドロップダウン リストからデータベースを選択します。

## パラメータ

## 説明

---

**Option.OutputFieldLevelReturnCodes**

フィールドレベルの結果インジケータを含めるかどうかを指定します。フィールドレベルの結果インジケータは、**ValidateAddressLocate** が各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、HouseNumber のフィールドレベルの結果インジケータは **HouseNumber.Result** に格納されます。結果インジケータの出力フィールドの完全な一覧は、[結果インジケータ](#) (704ページ) を参照してください。

- N** フィールドレベルのリターンコードを出力しません(デフォルト)。
  - Y** フィールドレベルのリターンコードを出力します。
-

## パラメータ

## 説明

## Option.OutputFormattedOnFail

住所の妥当性を確認できない場合に書式を整えた住所を返すかどうかを指定します。住所には、その国の標準住所書式が設定されます。このオプションを選択しない場合、**ValidateAddressLoqate** が住所の妥当性を確認できないと、出力住所フィールドは空白になります。

**N** 失敗した住所の書式を整えません (デフォルト)。

**Y** 失敗した住所の書式を整えます。

**[標準住所を含める]**、**[住所行の要素を含める]**、および **[郵便情報を含める]** の各チェックボックスで指定されたフォーマットを使用して、書式を整えた住所が返されます。**[住所行の要素を含める]** を選択した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。**ValidateAddressLoqate** が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**[正規化された入力住所要素を含める]** を選択します。

このオプションをオンにする場合は、**[標準住所を含める]** と **[住所行の要素を含める]** またはそのいずれかを選択する必要があります。

**OutputRecordType** オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。**OutputRecordType=E** を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。**ValidateAddressLoqate** が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**OutputRecordType=I** を指定します。

Y を指定する場合は、**OutputRecordType** に対して "A" と "E" またはそのいずれかを指定する必要があります。

**Option.OutputRecordType** オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。

**Option.OutputRecordType=E** を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。**ValidateAddressLoqate** が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**Option.OutputRecordType=I** を指定します。

Y を指定する場合は、**Option.OutputRecordType** に対して "A" と "E" またはそのいずれかを指定する必要があります。

## パラメータ

## 説明

---

**Option.OutputAddressBlocks**

実際の郵便物に印字される、書式を整えた住所を返すかどうかを指定します。住所の各行が、別々の住所ブロック フィールドに入れて返されます。**AddressBlock1** から **AddressBlock9** まで、最大 9 つの住所ブロック出力フィールドが使用されます。

例えば、以下の住所入力の場合、

AddressLine1: 4200 Parliament Place  
AddressLine2: Suite 600  
City: Lanham  
StateProvince: MD  
PostalCode: 20706

以下の住所ブロックが出力されます。

AddressBlock1: 4200 PARLIAMENT PL STE 600  
AddressBlock2: LANHAM MD 20706-1882  
AddressBlock3: UNITED STATES OF AMERICA

**ValidateAddressLoqate** は、郵便当局の規格に従って住所の書式を整え、住所ブロックの形式にします。国名は、万国郵便連合 (UPU) の国名で返されます。オプションは、住所ブロックの国名には影響を与えないことに注意してください。[国フォーマット]オプションは、**Country** 出力フィールドに返される名前だけに影響を与えます。

次のいずれかです。

- N** 住所ブロックを返しません。こちらがデフォルトです。
- Y** 住所ブロックを返します。



## パラメータ

## 説明

## Option.AmasFormatting

Address Matching Approval System (AMAS) 表記を使用して出力住所データをフォーマットすることを指定します。

このオプションを使用すると、Validate Address Loqate は、住所を正規化するとき AMAS ルールを使用するようになります。AMAS は、オーストラリア郵政公社が定める、住所規格を徹底するためのプログラムです。AMAS 書式設定表記の詳細については、『Address Matching Approval System (AMAS) Handbook』を参照してください。

このオプションを使用すると、出力データは次のように変更されません。

- 数値フィールドにはゼロが付加されます。この影響を受けるのは、HouseNumber、HouseNumber2、PostalDeliveryNumber、および DPID の各出力フィールドです。例えば、入力フィールドが 298 New South Head Rd Double Bay NSW 2028 の場合、HouseNumber フィールドの形式は 298 から 00298 に変更されます。
- 一致しない場合、DPID フィールドの桁はすべてゼロになります。例えば、00000000 などです。
- 一致しない場合、すべてゼロを含む数値フィールドを除き、すべてのリターン フィールド (パース済み住所要素) が空白になります。
- CCD フィールドは出力されません。

有効な値は、次のとおりです。

**N** AMAS 表記を使用して出力データをフォーマットしません (デフォルト)。

**Y** AMAS 表記を使用して出力データをフォーマットします。

注: このオプションを選択すると、[許容レベル] フィールドと [最小マッチ スコア] フィールドの選択に関わらず、AMAS フォーマットで結果が返されます。

## Option.OutputCasing

出力データの大文字と小文字の区別を指定します。次のいずれかです。

**M** 出力には、大文字と小文字が混在させます (デフォルト)。例:  
123 Main St  
Mytown FL 12345

**U** 出力に大文字を使用します。例:  
123 MAIN ST  
MYTOWN FL 12345

## パラメータ

## 説明

## Option.HomeCountry

デフォルト国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がドイツにある場合は、ドイツを指定します。ValidateAddressLoqate は、[StateProvince]、[PostalCode]、および [Country] の各住所フィールドから国を特定できなかった場合、指定された国を使用して、住所の検証を試みます。有効な国名には次のものがあります。

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

パラメータ	説明
Option.OutputCountryFormat	<p><b>Country</b> 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。</p> <p><b>E</b> 英語の国名を使用します (デフォルト)。</p> <p><b>I</b> 国名の代わりにその国の 2 文字の ISO の略語を使用します。</p> <p><b>U</b> 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。</p>
Option.OutputScript	<p>出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。</p> <p><b>Input</b> 書き直しを実行せず、入力と同じスクリプトで出力します (デフォルト)。</p> <p><b>Native</b> 使用可能な場合は、選択した国のネイティブスクリプトで出力します。</p> <p><b>Latn</b> 英語の値を使用します。</p>

パラメータ

説明

---

許容レベル

Option.AcceptanceLevel

## パラメータ

## 説明

レコードが正常に処理されたとみなされるために達成する必要のある、最小検証レベルを指定します。このフィールドの値は、“処理後検証マッチ レベル” と呼ばれる、Address Verification Code の 2 つめの文字に対応します。

- **5** — 配達ポイント (建物または郵便受け)。入力レコードの ApartmentNumber、HouseNumber、Street、City、StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。ApartmentNumber は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、ApartmentNumber は他のフィールドよりも細かいレベルであるため、Loqate エンジンはこの場合に、ApartmentNumber を特定できるはずでず。Loqate エンジンが ApartmentNumber とその他のフィールドをパーシングできない場合は、確信レベルは 0 となります。
- **4** — 敷地または建物。入力レコードの HouseNumber、Street、City、StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。HouseNumber は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、HouseNumber は他のフィールドよりも細かいレベルであるため、Loqate エンジンはこの場合に、HouseNumber を特定できるはずでず。Loqate エンジンが HouseNumber とその他のフィールドをパーシングできない場合は、確信レベルは 0 となります。
- **3** — 大通り、道路、または通り。入力レコードの Street、City、StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。City は正しいが StateProvince が正しくない場合、確信レベルは中程度になりますが、City は StateProvince に含まれるため、Loqate エンジンはこの場合に、StateProvince を特定できるはずでず。Loqate エンジンが City または両方のフィールド (City と StateProvince) をパーシングできない場合は、確信レベルは 0 となります。
- **2** — 地方 (都市または町)。入力レコードの City と StateProvince の両方が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。City は正しいが StateProvince が正しくない場合、確信レベルは中程度になりますが、City は StateProvince に含まれるため、Loqate エンジンはこの場合に、StateProvince を特定できるはずでず。Loqate エンジンが City または両方のフィールド (City と StateProvince) をパーシングできない場合は、確信レベルは 0 となります。
- **1** — 行政区域 (州または地域)。入力レコードの StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。
- **0** — なし。これは、最も制限の緩いマッチオプションと同じです。

パラメータ	説明
Option.IsDuplicateHandlingMaskEnable	<p>重複処理マスクを有効にし、重複レコードの処理および削除の方法を指定します。次のオプションから 1 つ以上を選択します。</p> <p><b>S</b> デフォルトで選択されています。入力の前処理により、単一フィールドで発生している重複を削除します。</p> <p><b>C</b> デフォルトで選択されています。入力の前処理により、すべてのフィールドわたり重複を削除します。</p> <p><b>T</b> 入力の前処理により、標準住所フィールドでないフィールド内の重複を削除します。</p> <p><b>F</b> デフォルトで選択されています。検証の出力の後処理により、検証されていないフィールドから重複を削除します。</p>
Option.MinimumMatchScore	<p>Validate Address Loqate において、Loqate 参照データベースでマッチ結果を得るために、住所に加える変更の度合いを、0～100 の間の数値で指定します。数値が小さいほど、大きな変更が許容されます。100 の値は、パーシング後に入力住所と検証済み住所がほぼ同一であることを意味します。0 の値は、検証済み住所を得るためにパーシング後の入力住所を大きく変更してしまってもよいことを意味します。</p>
Option.KeepMultimatch	<p>一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを指定します。</p> <p><b>Y</b> 複数のマッチを返します (デフォルト)。</p> <p><b>N</b> 複数のマッチを返しません。</p> <p>詳細については、<a href="#">複数マッチを返す</a> (694 ページ) を参照してください。</p>
Option.FailMultipleMatches	<p>一致する可能性のある住所が複数存在する入力住所に対して、複数の住所を許可しません。</p>

### 複数マッチを返す

ValidateAddressLoqate が、入力住所に一致する可能性のある複数の住所を郵便データベース内で検出した場合に、ValidateAddressLoqate がそれらの複数の住所を返すように設定できます。例えば、次の住所は米国郵便データベース内の複数の住所にマッチします。

PO BOX 1 New York, NY

## オプション

複数マッチを返すには、次の表に示すオプションを使用します。

表 59 : 複数マッチのオプション

### 説明/有効値

一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを示します。

返す住所の最大数を示す 1 ~ 10 の数字を入力します。デフォルト値は 1 です。

注：との違いは、は複数マッチによって失敗が返され、は複数マッチによって 1 つのレコードが返される点です。

どの出力住所が候補住所かを特定するには、必要があります。このように設定すると、候補住所のレコードのフィールドレベルの結果インジケータに 1 つ以上の値 "M" が格納されます。

## 出力

複数マッチを返すよう選択した場合、住所は指定した住所フォーマットで返されます。住所フォーマットの指定については、[オプション](#) (686ページ) を参照してください。どのレコードが候補住所であるかを特定するには、フィールドレベルの結果インジケータに複数の値 "M" があるかどうかを調べます。詳細については、[結果インジケータ](#) (704ページ) を参照してください。

### マッチ スコアのしきい値オプション

マッチ スコアのしきい値を設定するための 2 つのオプションがあります。

注：これらのオプションは `Validate Address Loqate` のユーザ インターフェイスにはなく、以下のファイルの中にあります。

```
SpectrumDirectory/server/modules/loqate/env.properties
```

**[MatchScoreAbsoluteThreshold]** オプションは、レコードがマッチングの候補とみなされるために達しなければならない最小マッチ スコアを指定するために使用されます。デフォルト値は 60 で、最大値は 100 です。

**[MatchScoreThresholdFactor]** は、最も高いマッチング結果を 100 とした場合の係数を表す値です。この値は、結果候補を検討する際のボーダーラインとして使用されます。係数の値が高いほど、良い検証結果が得られる確率が高くなります。デフォルト値は 95 で、最大値は 100 です。

## 応答

ValidateAddressLoqate からの出力には、選択した出力カテゴリに応じてさまざまな情報が含まれます。

### 標準住所出力

標準住所出力は、宛名ラベルに表記される住所に対応する 4 行の住所で構成されます。都市、州/省、郵便番号などのデータも、標準住所出力に含まれます。**[標準住所を含める]** チェックボックスを選択 **OutputRecordType = A** と設定した場合、ValidateAddressLoqate は、妥当性を確認した住所に対し、標準住所出力を返します。したかどうにかかわらず、妥当性が確認できなかった住所に対しては、標準住所フィールドが必ず返されます。妥当性が確認されなかった住所に対しては、標準住所出力フィールドには、入力住所がそのまま含まれます ("パス スルー" データ)。ValidateAddressLoqate において、妥当性が確認できなかった場合に、郵便当局の規格に従って住所を正規化するには、します。

表 60 : 標準住所出力

応答要素	説明
AdditionalInputData	特定の住所コンポーネントに一致しなかった入力データ。詳細については、 <a href="#">その他の入力データについて</a> を参照してください。
AddressLine1-4	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 1 行目です。住所の妥当性が確認できなかった場合は、入力住所の 1 行目がそのまま出力されます。住所ブロック出力フィールドは最大で、AddressLine1 から AddressLine4 の 4 つになります。
City	妥当性が確認された都市名。
Country	で選択した、以下のいずれかのフォーマットで示された国。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>



応答要素	説明
FirmName	妥当性が確認された企業名。
PostalCode	妥当性が確認された ZIP Code™ または郵便番号。
PostalCode.AddOn	ZIP Code™ の 4 桁のアドオン部分。例えば、60655-1844 という ZIP Code™ において、4 桁のアドオン部分は 1844 になります
PostalCode.Base	5 桁の ZIP Code™。例: 20706。
StateProvince	妥当性が確認された州または省の略称。

#### パース済み住所要素出力

した場合、出力住所は、パース済み住所の形式で書式設定されます。ValidateAddressLoqate で、妥当性が確認できなかった場合に、パース済み住所形式で書式設定されたデータ (正規化済み住所) を返すには、します。

**注:** ValidateAddressLoqate で、妥当性が確認できたかどうかにかかわらず、常にパースした入力データを返すには、します。詳細については、[パース済み入力](#) (700ページ) を参照してください。

表 61 : パース済み住所出力

Response Element	説明
AddressBlock1-9	<p>AddressBlock 出力フィールドには、正規化済み、または標準化済みの住所が、実際の郵便物に印刷される形式にフォーマットされて入ります。Validate Address Global は住所を郵便当局の規格に従って住所ブロックにフォーマットします。住所の各行が、別々の住所ブロックフィールドに入れて返されます。AddressBlock1 から AddressBlock9 まで、最大 9 つの住所ブロック出力フィールドが使用されます。例えば、以下の入力住所の場合、</p> <p>AddressLine1: 4200 Parliament Place AddressLine2: Suite 600 City: Lanham StateProvince: MD PostalCode: 20706</p> <p>以下の住所ブロックが出力されます。</p> <p>AddressBlock1: 4200 PARLIAMENT PL STE 600 AddressBlock2: LANHAM MD 20706-1882</p>
ApartmentLabel	<p>アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b></p>
ApartmentNumber	<p>アパート番号。例: 123 E Main St <b>APT 3</b></p>
ApartmentNumber2	<p>補助的なアパート番号。例: 123 E Main St <b>APT 3, 4th Floor</b></p> <p>注: このリリースでは、このフィールドは常に空白になります。</p>
Building	<p>個々の場所を識別するためのわかりやすい名前。</p>
City	<p>妥当性が確認された都市名</p>
Country	<p>国。フォーマットは、 で選択したものになります。</p> <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>

Response Element	説明
County*	国における最小の地理的データ要素。例: 米国の郡
FirmName	妥当性が確認された企業名
HouseNumber	家番号 1。例: <b>123</b> E Main St Apt 3
LeadingDirectional	接頭方向指示。例: 123 <b>E</b> Main St Apt 3
POBox	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode	妥当性が検証された郵便番号。米国住所に対しては、ZIP Code と呼びます。
Principality *	国における最大の地理的データ要素
StateProvince	妥当性が確認された州または省の名前
StreetAlias	通り名のエイリアス。通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。例: 123 <b>E Main</b> St Apt 3
StreetName	通り名。例: 123 <b>E Main St</b> Apt 3
StreetSuffix	通り接尾語。例: 123 E Main <b>St</b> Apt 3
Subcity*	[Locality] フィールドの内容に依存する、より小さな居留区データ要素。例: トルコ人居留区。

Response Element	説明
Substreet*	国における従属通りまたはブロック データ要素。例: 英国の従属通り。
TrailingDirectional	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>

\*これはサブフィールドであり、データを含まない場合があります。

### パース済み入力

出力には、パース済み形式で入力住所を含めることができます。このようなタイプの出力は、"パース済み入力" と呼ばれます。パース済み入力フィールドには、**ValidateAddress** が住所の妥当性を検証したかどうかにかかわらず、入力として使用される住所データが含まれます。パース済み入力は、住所の妥当性を検証できた場合にパース済み住所要素に妥当性が検証された住所が含まれ、オプションで、住所の妥当性が検証できなかった場合には入力データが含まれるという点で、"パース済み住所要素" 出力と異なります。パース済み入力には、**ValidateAddress** が住所の妥当性を検証したかどうかにかかわらず、常に入力住所が含まれます。

パース済み入力フィールドを出力に含めるには、。

表 62 : パース済み入力

応答要素	説明
ApartmentLabel.Input	アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b>
ApartmentNumber.Input	アパート番号。例: 123 E Main St <b>APT 3</b>
City.Input	妥当性が確認された都市名
Country.Input	国。フォーマットは、 で選択したものになります。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>

応答要素	説明
County.Input*	国における最小の地理的データ要素。例: 米国の郡
FirmName.Input	妥当性が確認された企業名
HouseNumber.Input	家番号 1。例: <b>123</b> E Main St Apt 3
LeadingDirectional.Input	接頭方向指示。例: 123 <b>E</b> Main St Apt 3
POBox.Input	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode.Input	妥当性が検証された郵便番号。米国住所に対しては、ZIP Code と呼びます。
Principality.Input *	国における最大の地理的データ要素
StateProvince.Input	妥当性が確認された州または省の名前
StreetAlias.Input	通り名のエイリアス。通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。基本名とは、通り全体に適用される名称です。例えば、StreetName が "N MAIN ST" の場合、StreetAlias フィールドには "MAIN" が含まれ、StreetSuffix フィールドには大通りタイプである "ST" が返されます。
StreetName.Input	通り名。例: 123 <b>E Main St</b> Apt 3
StreetSuffix.Input	通り接尾語。例: 123 E Main St <b>Apt 3</b>

応答要素	説明
Subcity.Input*	[Locality] フィールドの内容に依存する、より小さな居留区データ要素。例: トルコ人居留区。
Substreet.Input*	国における従属通りまたはブロック データ要素。例: 英国の従属通り。
TrailingDirectional.Input	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>

\*これはサブフィールドであり、データを含まない場合があります。

#### ジオコード出力

`ValidateAddressLoqate` は、緯度/経度、ジオコーディング マッチ コード、従属する地方、従属地方に含まれる地方、従属する大ストリート、下位行政区画と上位行政区画、検索距離を出力として返します。マッチ コードは、入力住所がどの程度まで既知の住所に一致したかを表すとともに、マッチングの全体的なステータスを示します。検索距離コードは、ジオコードが住所の実際の物理的な位置にどの程度近いかを表します。

表 63 : ジオコード住所出力

応答要素	説明
Geocode.MatchCode	<p>住所に対するジオコード マッチングのステータスとレベルを表す 2 バイト コード。</p> <p>最初のバイトはジオコーディング ステータスを表し、次のいずれかになります。</p> <p><b>A</b> 入力住所に一致する複数のジオコード候補が検出され、それらの平均が返されました。</p> <p><b>I</b> ある範囲の入力住所のロケーションからジオコードを補間することができました。</p> <p><b>P</b> 入力住所に一致する単一のジオコードが検出されました。</p> <p><b>U</b> 入力住所に対するジオコードを生成できませんでした。</p> <p>2 つめのバイトはジオコード マッチングのレベルを表し、次のいずれかになります。</p> <p><b>5</b> 配達ポイント (郵便受けまたは従属する建物)</p> <p><b>4</b> 敷地または建物</p> <p><b>3</b> Thoroughfare</p> <p><b>2</b> Locality</p> <p><b>1</b> 行政区画</p> <p><b>0</b> なし</p>
Latitude	小数第 5 位まで計算される 8 桁の度数 (指定したフォーマットで表記されます)。
Longitude	小数第 5 位まで計算される 8 桁の度数 (指定したフォーマットで表記されます)。
SearchDistance	メートル単位で表した精度の半径。与えられたジオコードと実際の物理的位置との間の推測最大距離を表します。このフィールドは、基盤の参照データから導き出され、その精度と対象範囲に依存します。

表 64 : 都市/通り/郵便番号セントロイド マッチ コード

要素	マッチ コード
住所ポイント	P4
住所ポイント補間済み	I4
通りセントロイド	A4/P3
郵便番号/都市セントロイド	A3/P2/A2

注 : `Geocode.Match.Code` は、通り セグメントに対して 2 つの座標を返すことはしません (通りの一部の開始と終了など)。代わりに、入力に対するリターンコードが I3 になった (大通りまたは通り レベルに補間され、敷地番号は入力されていない) 場合は、通り全体が計算に使用されます。

#### 結果インジケータ

結果インジケータは、住所に対して実行した処理の種類に関する情報を提供します。結果インジケータには、次の 2 種類があります。

#### レコード レベルの結果インジケータ

レコード レベルの結果インジケータは、各レコードに対する `ValidateAddressLoqate` 処理の結果に関するデータを提供します。例えば、マッチングの成功または失敗、住所を処理したコーダーなどの詳細情報を示します。以下の表に、`ValidateAddressLoqate` が返すレコード レベルの結果インジケータの一覧を示します。

表 65 : レコード レベル インジケータ

応答要素	説明
Confidence	返された住所に割り当てられた確信レベル。範囲は 0 ~ 100 です。0 は失敗を表し、100 はマッチ結果が正しいことに対する確信レベルが非常に高いことを表します。複数のマッチ結果がある場合、確信レベルは 0 です。この数値の計算方法については、 <a href="#">Validate Address Loqate 確信アルゴリズムの概要</a> を参照してください。



## 応答要素

## 説明

**CouldNotValidate**

マッチしなかった場合の、妥当性が確認できなかった住所コンポーネント。

- ApartmentNumber
- HouseNumber
- StreetName
- PostalCode
- City
- Directional
- StreetSuffix
- Firm
- POBoxNumber

注：複数のコンポーネントがカンマ区切りリストとして返されることがあります。

**MatchScore**

**MatchScore** は、入力データと、最も近い一致として検出された参照データとの間の類似性を表します。**MatchScore** は、一致を得るために入力住所をどれだけ変更したかを表す **Confidence** とはまったく異なるものです。**MatchScore** の意味は、米国住所と米国以外の住所で異なります。

`int getFieldMatchscore (unit record, const char*)` 関数は、入力データと、最も近い一致として検出された参照データとの間の類似性を表す 0 ~ 100 の整数値です。100 は、入力データに、エイリアス、大文字小文字、付加記号以外の変更を加えなかったことを表します。0 は、入力データと、最も近い一致として検出された参照データとの間にまったく類似性がないことを表します。

注：**Validate Address Loqate** および **Advanced Matching** モジュールのコンポーネントは、どちらも **MatchScore** フィールドを使用します。データフローの出力の **MatchScore** フィールドの値は、出力ステージに送られる前に最後に値を変更したステージによって決まります。データフローに **Validate Address Loqate** および **Advanced Matching** モジュールのコンポーネントが含まれ、各ステージの **MatchScore** 出力フィールドを確認したい場合は、**Transformer** ステージを使用して、**MatchScore** 値を他のフィールドにコピーしてください。例えば、**Validate Address Loqate** によって **MatchScore** という出力フィールドが作成され、**Transformer** ステージによって **Validate Address Loqate** の **MatchScore** フィールドが **AddressMatchScore** というフィールドにコピーされます。マッチャーステージを実行すると、マッチャーから得た値が **MatchScore** フィールドに設定され、**Validate Address Loqate** から得た **AddressMatchScore** の値が引き渡されます。

**ProcessedBy**

住所を処理した住所コーダーです。

**LOQATE**

Loqate コーダーが住所を処理しました。

応答要素	説明				
Status	<p>マッチの成功または失敗。複数のマッチがある場合、一致する可能性のあるすべてのものに対してこのフィールドが "F" になります。</p> <table border="1"> <tr> <td>NULL</td> <td>成功</td> </tr> <tr> <td>F</td> <td>失敗</td> </tr> </table>	NULL	成功	F	失敗
NULL	成功				
F	失敗				
Status.Code	<p>失敗の原因 (ある場合)。</p> <ul style="list-style-type: none"> <li>UnableToValidate</li> </ul>				
Status.Description	<p>問題の説明 (ある場合)。</p> <p><b>Address Not Found</b> Status.Code=UnableToValidate の場合にこの値が表示されます。</p>				

### フィールドレベルの結果インジケータ

フィールドレベルの結果インジケータは、`ValidateAddressLoqate` が各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、`HouseNumber` のフィールドレベルの結果インジケータは **HouseNumber.Result** に格納されます。

フィールドレベルの結果インジケータを有効にするには、。

次の表に、フィールドレベルの結果インジケータの一覧を示します。特定のフィールドが住所に適用されない場合、結果インジケータが空白になる場合があります。

表 66 : フィールドレベルの結果インジケータ

応答要素	説明
ApartmentLabel.Result	<b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
	<b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。
	<b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。
	<b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
	<b>R</b> アパート ラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
	<b>S</b> 正規化。このオプションには、標準の略語が含まれます。
	<b>U</b> マッチしない。カナダの住所には適用されません。
<b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。	
ApartmentNumber.Result	<b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
	<b>C</b> 修正済み。カナダの住所のみ。
	<b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
	<b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国EWSにマッチする米国の住所には、Pの値が割り当てられます。米国およびカナダの住所のみをサポートします。
	<b>R</b> アパート番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
	<b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
	<b>U</b> マッチしない。
<b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。	

応答要素	説明
City.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> ハイフンの欠落または句読文字エラー。カナダの住所のみ。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。</p> <p><b>R</b> 都市名が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。</p> <p><b>U</b> マッチしない。カナダの住所には適用されません。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
Country.Result	<p>これらの結果コードは、米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
County.Result*	<p>国における最小の地理的データ要素。例: 米国の郡</p>

## 応答要素

## 説明

FirmName.Result

- C** 修正済み。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
- U** マッチしない。米国およびカナダの住所のみをサポートします。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。米国住所にのみ適用されます。

HouseNumber.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
- C** 修正済み。カナダの住所のみ。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- O** 範囲外。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
- R** 家番号が必須ですが、入力住所から欠落しています。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

応答要素	説明
LeadingDirectional.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。空白のない入力が、空白のない値に修正されました。米国住所にのみ適用されます。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした、出力に保持されました。カナダの住所のみ。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。カナダの住所には適用されません。</p>
POBox.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。</p> <p><b>C</b> 修正済み。カナダの住所のみ。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数マッチ。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした、出力に保持されました。カナダの住所のみ。</p> <p><b>R</b> 私書箱番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>

応答要素	説明
PostalCode.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。カナダの住所には適用されません。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした、出力に保持されました。米国住所には適用されません。</p> <p><b>R</b> 郵便番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。</p> <p><b>U</b> マッチしない。例えば、通り名と郵便番号が一致しない場合、<b>StreetName.Result</b> と <b>PostalCode.Result</b> の両方に <b>U</b> が割り当てられます。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
PostalCode.Type	<p><b>P</b> ZIP Code™ には、PO Box 住所のみが含まれます。米国住所にのみ適用されます。</p> <p><b>U</b> ZIP Code™ は、特定の会社または場所に割り当てられたユニークな ZIP Code™ です。米国住所にのみ適用されます。</p> <p><b>M</b> ZIP Code™ は、軍施設の住所です。米国住所にのみ適用されます。</p> <p><b>NULL</b> ZIP Code™ は、標準 ZIP Code™ です。</p>
Principality.Result *	国における最大の地理的データ要素

## 応答要素

## 説明

**StateProvince.Result**

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国住所にのみ適用されます。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
- R** アパート ラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

**StreetAlias.Result**

通り名のエイリアス。通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。基本名とは、通り全体に適用される名称です。例えば、**StreetName** が "N MAIN ST" の場合、**StreetAlias** フィールドには "MAIN" が含まれ、**StreetSuffix** フィールドには大通りタイプである "ST" が返されます。



応答要素	説明
StreetName.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国住所には適用されません。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国およびカナダの住所のみをサポートします。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
StreetSuffix.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。米国住所には適用されません。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
Subcity.Result*	[Locality] フィールドの内容に依存する、より小さな居留区データ要素。例: トルコ 人居留区。

## 応答要素

## 説明

Substreet.Result*	国における従属通りまたはブロック データ要素。例: 英国の従属通り。
TrailingDirectional.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。カナダの住所には適用されません。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>

\*これはサブフィールドであり、データを含まない場合があります。

## AVC コード

Address Verification Code (AVC) は、住所に対する精度インジケータで構成される 11 バイトのコードです。これらのコードは、処理結果の品質を表し、必要に応じて入力データを修正する方法に関するガイドラインを示します。個別住所のそれぞれに、独自のコードが付与されます。このコードは、データフローの出力内で自動的に返されます。AVC は、以下のような形式です。

V44-I44-P6-100

AVC は、8 つの部分で構成されます。

- 検証ステータス
- 処理後検証マッチ レベル
- 処理前検証マッチ レベル
- パーシング ステータス
- 辞書識別マッチ レベル
- コンテキスト識別マッチ レベル
- 郵便番号ステータス

- マッチスコア

### 検証ステータス

住所の検証レベル。

- **V** — 検証済み。入力データは、使用可能な参照データからの単一のレコードと完全に一致しています。シンプルな住所検証では、このコードが返されることが最良の結果とみなされます。
- **P** — 部分的に検証済み。入力データは、使用可能な参照データからの単一のレコードと部分的に一致しています。住所情報の詳細なデータが提供されているが、完全な検証を行うためには追加情報が必要であることを意味する可能性があります。
- **A** — 曖昧。入力に一致する可能性のある複数の住所があります。
- **U** — 検証不可。住所を検証するための十分な情報がない場合や、入力クエリが読み取れない場合に、これが返されます。出力フィールドには入力データが含まれます。
- **R** — 元に戻されました。レコードを、指定された最小許容レベルで検証できませんでした。元に戻すための最小レベルなどの詳細設定オプションが、処理において設定されている場合に生じます。出力フィールドには入力データが含まれます。
- **C** — 矛盾。相反する値を持つ複数の参照データと近似一致します。

### 処理後検証マッチ レベル

使用可能な参照データに対する、入力データの処理後のマッチ レベル。

- **5** — 配達ポイント (建物または郵便受け)。入力レコードの `ApartmentNumber`、`HouseNumber`、`Street`、`City`、`StateProvince` が、`Loqate` のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。`ApartmentNumber` は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、`ApartmentNumber` は他のフィールドよりも細かいレベルであるため、`Loqate` エンジンはこの場合に、`ApartmentNumber` を特定できるはずです。`Loqate` エンジンが `ApartmentNumber` とその他のフィールドをパーシングできない場合は、確信レベルは **0** となります。
- **4** — 敷地または建物。入力レコードの `HouseNumber`、`Street`、`City`、`StateProvince` が、`Loqate` のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。`HouseNumber` は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、`HouseNumber` は他のフィールドよりも細かいレベルであるため、`Loqate` エンジンはこの場合に、`HouseNumber` を特定できるはずです。`Loqate` エンジンが `HouseNumber` とその他のフィールドをパーシングできない場合は、確信レベルは **0** となります。
- **3** — 大通り、道路、または通り。入力レコードの `Street`、`City`、`StateProvince` が、`Loqate` のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。`City` は正しいが `StateProvince` が正しくない場合、確信レベルは中程度になりますが、`City` は `StateProvince` に含まれるため、`Loqate` エンジンはこの場合に、`StateProvince` を特定できるはずです。`Loqate` エンジンが `City` または両方のフィールド (`City` と `StateProvince`) をパーシングできない場合は、確信レベルは **0** となります。

- **2** — 地方 (都市または町)。入力レコードの **City** と **StateProvince** の両方が、**Loqate** のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。**City** は正しいが **StateProvince** が正しくない場合、確信レベルは中程度になりますが、**City** は **StateProvince** に含まれるため、**Loqate** エンジンはこの場合に、**StateProvince** を特定できるはずですが。**Loqate** エンジンが **City** または両方のフィールド (**City** と **StateProvince**) をパーシングできない場合は、確信レベルは **0** となります。
- **1** — 行政区域 (州または地域)。入力レコードの **StateProvince** が、**Loqate** のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。
- **0** — なし。これは、最も制限の緩いマッチ オプションと同じです。

### 処理前検証マッチ レベル

使用可能な参照データに対する、入力データの処理前のマッチ レベル。

- **5** — 配達ポイント (建物または郵便受け)
- **4** — 敷地または建物。
- **3** — 大通り、道路、または通り。
- **2** — 地方 (都市または町)。
- **1** — 行政区域 (州または地域)。
- **0** — なし。

### パーシング ステータス

住所のパーシング レベル。

- **I** — 識別およびパーシング済み。入力データは識別され、各コンポーネントに配置されました。例えば、"123 Kingston Av" に対して **Validate Address Loqate** は、"123" が敷地番号で、"Kingston" が大通り名で、"Av" または "Avenue" が大通りのタイプであると判断できます。
- **U** — パーシング不可。**Validate Address Loqate** は、入力データを識別およびパーシングできませんでした。"未検証"の検証ステータスと同様に、入力データは不完全またはあいまいでした。

### 辞書識別マッチ レベル

パターン マッチング (例えば、数値は敷地番号である可能性があります) や辞書マッチング (例えば、"rd" は大通りのタイプ "road" であったり、"London" は地方であったりする可能性があります) の適用による、入力データの認識済み形式のレベル。

- **5** — 配達ポイント (建物または郵便受け)
- **4** — 敷地または建物。
- **3** — 大通り、道路、または通り。
- **2** — 地方 (都市または町)。
- **1** — 行政区域 (州または地域)。
- **0** — なし。

### コンテキスト識別マッチ レベル

出現のコンテキストに基づいて認識可能な、入力データのレベル。最も精度の低い形式のマッチングで、単語を特定の住所要素として識別することに基づいて行われます。例えば、前に敷地らしい要素があり、後に地方らしい要素が続き、後の項目は参照データまたは辞書とのマッチングによって識別されている場合、入力は大通りであると判断できる可能性があります。

- **5** — 配達ポイント (建物または郵便受け)
- **4** — 敷地または建物。
- **3** — 大通り、道路、または通り。
- **2** — 地方 (都市または町)。
- **1** — 行政区域 (州または地域)。
- **0** — なし。

### 郵便番号ステータス

郵便番号の検証レベル。

- **P8** — PostalCodePrimary と PostalCodeSecondary が検証済みです。
- **P7** — PostalCodePrimary は検証済みで、PostalCodeSecondary は追加または変更されています。
- **P6** — PostalCodePrimary が検証済みです。
- **P5** — PostalCodePrimary が、小さな変更を加えることによって検証済みです。
- **P4** — PostalCodePrimary が、大きな変更を加えることによって検証済みです。
- **P3** — PostalCodePrimary が追加されています。
- **P2** — PostalCodePrimary が辞書によって識別されています。
- **P1** — PostalCodePrimary がコンテキストによって識別されています。
- **P0** — PostalCodePrimary が空です。

### マッチ スコア

識別済みの入力データと、レコードの出力データの間類似性を表す 0 ~ 100 の間の数値。100 は、入力データに追加、エイリアス、大文字小文字、付加記号以外の変更を加えなかったことを表します。0 は、入力データと出力データの間まったく類似性がないことを表します。

### AMAS 出力

次の表に、ValidateAddressAUS が出力する標準フィールドを示します。

表 67 : 出力フィールド

応答要素	説明
Barcode	DPID に基づく標準バーコード。 <b>F</b> 失敗 (バーコード不検出) <b>20 桁の数字</b> 成功
DPID	配達ポイント識別子。通り住所などの郵便物配達ポイントを一意に識別する 8 桁の数字。オーストラリア郵政公社郵便住所ファイルに規定されています。 注：このフィールドは、AMAS で検証されていないオーストラリアの住所の場合は "00000000" となり、オーストラリア以外の住所の場合は空になります。
FloorNumber	フロアルevel番号。例: 123 E Main St Apt 3, <b>4th Floor</b>
FloorType	フロアルevel タイプ。例: 123 E Main St Apt 3, <b>4th Floor</b>
PostalBoxNum	郵便配達番号。例: PO Box 42

## Universal Name モジュール

### OpenNameParser

OpenNameParser は、名前データフィールドにある個人名、企業名、またはその他の名称を構成要素に分解します。これらのパースされた名前要素は、名前のマッチング、名前の正規化、複数レコード名の統合など、他の自動化処理に使用できます。

OpenNameParser は、次の処理を行います。

- 名前が担う機能を示すために、その名前のタイプを特定します。名前エンティティタイプは、個人名と企業名の 2 つのグループに分かれます。それぞれのグループには、さらに複数のサブグループがあります。

- パーシングに使う構文を把握するために、名前の形式を特定します。個人名は、通常、自然な (署名) 順序または逆の順序に従います。企業名は、通常、階層型の順序に従います。
- 名前を構成する各要素が名前全体に占める構文上の関連性を識別するために、要素を特定してラベル付けします。個人名の構文は、敬称、名、ミドルネーム、姓、接尾語、アカウントを示す用語、その他の個人名要素で構成されます。企業名の構文は、企業名や接尾語などで構成されます。
- 結合された個人名と企業名をパースし、それらを 1 つのレコードとして残すか、複数のレコードに分割します。Examples of conjoined names include "Mr.and Mrs.John Smith" and "Baltimore Gas & Electric dba Constellation Energy".
- 出力をレコードまたはリストとしてパースします。
- パーシングによる訂正の信頼度を示すパーシング スコアを割り当てます。

### リソース URL

JSON エンドポイント:

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

XML エンドポイント:

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

### JSON 応答の例

次の例では、JSON 応答を要求します。

```
http://myserver:8080/rest/OpenNameParser/results.json?
Data.Name=John+Williams+Smith
```

この要求から次のような JSON が返されます。

```
{ "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": []
} ] }
```



### XML 応答の例

次の例では、XML 応答を要求します。

```
http://myserver:8080/rest/OpenNameParser/results.xml?  
Data.Name=John+Williams+Smith
```

この要求から次のような XML が返されます。

```
<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>
```



## 要求

### 入力データのパラメータ

表 68 : Open Name Parser の入力

パラメータ	説明
Data.CultureCode	<p>入力された名前データのカルチャー。オプションは次のとおりです。</p> <p><b>Null (empty)</b>      グローバル カルチャー (デフォルト)。</p> <p><b>de</b>                      ドイツ語。</p> <p><b>es</b>                      スペイン語。</p> <p><b>ja</b>                      日本語。</p> <p>注 : Open Parser ドメインエディタを使用して独自のドメインを追加した場合、そのドメインのカルチャーとカルチャー コードも有効になります。</p>
Data.Name	<p>パースしたい名前。このフィールドは必須です。</p>

## オプション

### パーシング オプションのパラメータ

次の表に、名前のパーシングを制御するオプションを示します。

表 69 : Open Name Parser パーシング オプション

パラメータ	説明
Option.ParseNaturalOrderPersonalNames	<p>敬称、名、ミドル ネーム、姓、および接尾語の順序で名前をパースするかどうかを指定します。</p> <p><b>true</b>                      正順序の個人名をパースします。</p> <p><b>false</b>                      正順序の名前をパースしません。</p>

パラメータ	説明
Option.ParseReverseOrderPersonalNames	<p>姓が最初に指定されている名前をパースするかどうかを指定します。</p> <p><b>true</b> 逆順序の個人名をパースします。</p> <p><b>false</b> 逆順序の名前をパースしません。</p>
Option.ParseConjoinedNames	<p>結合名をパースするかどうかを指定します。</p> <p><b>true</b> 結合名をパースします。</p> <p><b>false</b> 結合名をパースしません。</p>
Option.SplitConjoinedNames	<p>Bill &amp; Sally Smith など、複数の人物を含む結合名を複数のレコードに分割するかどうかを指定します。</p> <p><b>true</b> 結合名を分割します。</p> <p><b>false</b> 結合名を分割しません。</p>
Option.ParseBusinessNames	<p>企業名をパースするかどうかを指定します。</p> <p><b>true</b> 企業名をパースします。</p> <p><b>false</b> 企業名をパースしません。</p>
Option.OutputAsList	<p>パース済み名前要素をリスト形式で返すかどうかを指定します。</p> <p><b>true</b> パース済み要素をリスト形式で返します。</p> <p><b>false</b> パース済み要素をリスト形式で返しません。</p>
Option.ShortcutThreshold	<p>パフォーマンスと品質のバランスをとる方法を指定します。パフォーマンスを上げると、品質出力が下がります。同様に、品質を上げると、パフォーマンスが下がります。このしきい値を満たすと、レコードに対して他の処理は実行されません。</p> <p>0 ~ 100 の値を指定します。デフォルト値は 100 です。</p>

## カルチャー オプションのパラメータ

次の表に、名前カルチャーを制御するオプションを示します。

**表 70 : Open Name Parser カルチャー オプション**

パラメータ	説明
Option.DefaultCulture	<p>パーシング グラマーに含めるカルチャーを指定します。デフォルトでは、グローバルカルチャーが選択されます。</p> <p>カンマ区切りリストに 2 文字のカルチャー コードを優先する順に指定することで、カルチャーを指定します。例えば、まずはスペインのカルチャー、次に日本のカルチャーを使用して名前のパースを試みるには、次のように指定します。</p> <pre>es, ja,,</pre>

#### 詳細オプションのパラメータ

次の表に、名前パーシング用の詳細オプションを示します。

**表 71 : Open Name Parser の詳細オプション**

オプション	説明
Option.NaturalOrderPersonalNamesDomain	<p>正順序個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。</p>
Option.NaturalOrderPersonalNamesPriority	<p>使用する他のドメインに対する正順序個人名ドメインの優先度を示す 1～5 の数値を指定します。これにより、実行するパーサーの順序が決定されます。</p> <p>ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。</p>

## オプション

## 説明

---

Option.ReverseOrderPersonalNamesDomain	逆順序個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。
Option.ReverseOrderPersonalNamesPriority	使用する他のドメインに対する逆順序個人名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。  ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。
Option.NaturalOrderConjoinedPersonalNamesDomain	正順序結合個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。
Option.NaturalOrderConjoinedPersonalNamesPriority	使用する他のドメインに対する正順序結合個人名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。  ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。
Option.ReverseOrderConjoinedPersonalNamesDomain	逆順序結合個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。

---

## オプション

## 説明

Option.ReverseOrderConjoinedPersonalNamesPriority	<p>使用する他のドメインに対する逆順序結合個人名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。</p> <p>ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。</p>
Option.BusinessNamesDomain	<p>企業名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。</p>
Option.BusinessNamesPriority	<p>使用する他のドメインに対する企業名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。</p> <p>ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。</p>

## 応答

表 72 : Open Name Parser の出力

応答要素	書式	説明
AccountDescription	文字列	名前の一部であるアカウント説明。例えば、"Mary Jones Account # 12345" で、アカウント説明は "Account#12345"。

応答要素	書式	説明
Names	文字列	パース済み要素のリストを含む階層フィールド。このフィールドは、[パーシングオプション]の[結果をリストに出力]ボックスをチェックしている場合に返されます。
会社名関係のフィールド		
FirmConjunction	文字列	"d/b/a" (doing business as)、"o/a" (operating as)、"t/a" (trading as) などの略語を含む企業の名前を示します。
FirmName	文字列	会社名。例えば、"Pitney Bowes"。
FirmSuffix	文字列	会社名の接尾語。例えば、"Co."、"Inc."
IsFirm	文字列	名前が、個人名ではなく、企業名であることを示します。
個人名に関するフィールド		
Conjunction	文字列	名前に、"and"、"or"、"&" などの接続詞が含まれることを示します。
CultureCode	文字列	入力データに含まれるカルチャー コード。
CultureCodeUsedToParse	文字列	データのパーズに使用されたカルチャー固有の文法を特定します。 <b>Null (empty)</b> グローバル カルチャー (デフォルト)。 <b>de</b> ドイツ語。 <b>es</b> スペイン語。 <b>ja</b> 日本語。  注：Open Parser ドメインエディタを使用して独自のドメインを追加した場合、そのドメインのカルチャーとカルチャーコードもこのフィールドに表示されます。

応答要素	書式	説明
FirstName	文字列	個人のファースト ネーム。
GeneralSuffix	文字列	個人名の一般/職業接尾語。例えば、 MD PhD 。
IsParsed	文字列	出力レコードがパースされたかどうかを示します。値は True または False です。
IsPersonal	文字列	名前が企業名ではなく、個人名であるかどうかを示します。値は True または False です。
IsReverseOrder	文字列	入力名が逆順序であるかどうかを示します。値は True または False です。
LastName	文字列	個人名のラスト ネーム。父方の姓が含まれます。
LeadingData	文字列	名前の前に付けられる、名前以外の情報。
MaturitySuffix	文字列	個人の世代/家族接尾語。例えば、 Jr.または Sr.。
MiddleName	文字列	個人のミドル ネーム。
Name.	文字列	入力に指定された個人名または企業名。
NameScore	文字列	各名前の既知および不明トークンの平均スコアを示します。NameScoreの値は、パーシング グラマーでの定義に従って、0 ~ 100 の間になります。マッチが返されない場合は、0 が返されます。
SecondaryLastName	文字列	スペイン語のパーシング グラマーでは、その人の母の姓。

応答要素	書式	説明
TitleOfRespect	文字列	"Mr.", "Mrs.", "Dr." など、名前の前に付けられる情報。
TrailingData	文字列	名名の後に付けられる、名前以外の情報。
結合名関係のフィールド		
Conjunction2	文字列	結合されている 2 番目の名前に、"and"、"or"、"&" などの接続詞が含まれることを示します。
Conjunction3	文字列	結合されている 3 番目の名前に、"and"、"or"、"&" などの接続詞が含まれることを示します。
FirmName2	文字列	結合されている 2 番目の企業名。例えば、Baltimore Gas & Electric dba Constellation Energy。
FirmSuffix2	文字列	結合されている 2 番目の企業の接尾語。
FirstName2	文字列	結合されている <b>FirstName</b> の 2 番目の名
FirstName3	文字列	結合されている <b>FirstName</b> の 3 番目の名。
GeneralSuffix2	文字列	結合されている 2 番目の名前の一般/職業接尾語。例えば、MD PhD 。
GeneralSuffix3	文字列	結合されている 3 番目の名前の一般/職業接尾語。例えば、MD PhD 。
IsConjoined	文字列	入力名が結合名であることを示します。結合名は、例えば、"John and Jane Smith"。



応答要素	書式	説明
LastName2	文字列	結合されている LastName の 2 番目の姓。
LastName3	文字列	結合されている LastName の 3 番目の姓。
MaturitySuffix2	文字列	結合されている 2 番目の名前の世代/家族接尾語。例えば、Jr.または Sr。
MaturitySuffix3	文字列	結合されている 3 番目の名前の世代/家族接尾語。例えば、Jr.または Sr。
MiddleName2	文字列	結合されている MiddleName の 2 番目の名。
MiddleName3	文字列	結合されている MiddleName の 3 番目の名。
TitleOfRespect2	文字列	"Mr."、"Mrs."、"Dr." など、結合されている 2 番目の名前の前に付けられる情報。
TitleOfRespect3	文字列	"Mr."、"Mrs."、"Dr." など、結合されている 3 番目の名前の前に付けられる情報。

# SOAP

## GeoConfidence モジュール

### GeoConfidence Surface

GeoConfidenceSurface は、Enterprise Geocoding モジュールによって生成されるジオコード情報の品質に基づいて地理信頼性ポリゴン（等高線とも呼ばれます）を返します。地理信頼性ポリゴンが生成されたら、そのポリゴンに他の空間データを重ねてリスクや確率を明らかにできます。

このサービスは、GeoConfidence モジュールの FloodZoneAnalysis データフロー テンプレートによって使用されます。

注：GeoConfidence では、Enterprise Geocoding モジュールおよび Spatial モジュールから提供されるサービスが利用されます。

### リソース URL

```
http://server:port/soap/GeoConfidenceSurface
```

### リクエスト

GeoConfidence Surface の入力フィールドは、Enterprise Geocoding モジュールの GeoConfidence 出力カテゴリから返された出力フィールドです。これらのフィールドについて、以下に詳しく説明します。

columnName フィールド名 レスポンス要素	最大null	説明
GeoConfidenceCode	13	<p>このフィールドに返される値は、どのタイプの Geoconfidence Surface が返されたかを示します。</p> <p>有効な値を次に示します。</p> <p><b>INTERSECTION</b> 2つの通りが交差する位置のジオコード ポイント。</p> <p><b>ADDRESS</b> 住所が位置付けられた通りセグメントを表す、通りセグメント ポイントの配列。</p> <p><b>POINT</b> ジオコードがポイント データを使って住所のマッチングに成功した場合、その住所が位置付けられたポイント ジオメトリ。</p> <p><b>POSTAL1</b> ZIP セントロイドのジオコード ポイント。</p> <p><b>POSTAL2</b> 住所が位置付けられた ZIP + 2 に含まれるすべての通りセグメントのポイント配列。</p> <p><b>POSTAL3</b> 住所が位置付けられた ZIP + 4 に含まれる通りセグメントのポイント配列。</p> <p><b>ERROR</b> エラーが発生しました。</p>
StreetSegmentPoints	1024	<p>通りセグメント ポイントを表す緯度/経度値の配列。</p> <p>注：このフィールドには、GeoConfidenceCode フィールドが ADDRESS,POSTAL2、または POSTAL3。</p>
GeoConfidenceCentroidLatitude	11	Geoconfidence ポリゴンのセントロイドの緯度。
GeoConfidenceCentroidLongitude	12	Geoconfidence ポリゴンのセントロイドの経度。

## 応答

[GeoConfidenceSurface] 出力フィールドには、Geoconfidence ポリゴンが格納されます。

応答要素	説明
Geometry	返されたジオメトリを表す Geoconfidence ポリゴン。

## Global Sentry モジュール

### GlobalSentry

GlobalSentry サービスは、各国から得られたデータによる政府提供のブラックリストと照合してトランザクション (取引) のマッチングを確認します。以下のリストが使用されます。

- Denied Persons List (米国)
- Unverified List (BIS Red Flag) (米国)
- Consolidated Financial Sanction Targets (Individuals and Entities) (英国/欧州連合)
- Consolidated lists of persons, groups, and entities subject to EU financial sanctions (欧州連合)
- DFAT Consolidated List (オーストラリア)
- OSFI Consolidated List (Individuals and Entities) (カナダ)
- Specially Designated Nationals, Terrorists, Narcotic Traffickers and other Blocked Persons List (米国)
- Statutorily Debarred Parties List (米国)
- Politically Exposed Persons (PEP) list
- 国連安全保障理事会が制裁対象に指定した個人および団体が含まれる統合リスト。

認可国、名前、住所、ID 番号、その他、誕生日などの情報と照合してマッチングを実行することにより、"総合的なリスク水準のスコア"が生成され、この情報に基づいて組織は特定の取引(トランザクション)を阻止すべきかどうかを的確に判断し、誤検出を回避できます。

GlobalSentry によるデータ処理の手順を以下に説明します。

1. Global Sentry サービス (以下、単に "サービス") は、最初にトランザクション内のすべての必須データをスキャンして制裁対象国を特定します。制裁対象国の一致が見つかったトランザクションについては、他のすべての検索条件がバイパスされ、最大級のリスク スコアが設定されます。

2. 制裁対象国の一致が見つからなかった場合、サービスは **GlobalSentry Name Check**、**GlobalSentry Address Check**、または **GlobalSentry ID Number Check** サブフローを使用して **GlobalSentry** データベースとの照合によるトランザクションのマッチングを試みます。
3. **GlobalSentry Name Check** は、個人、団体、および船舶に対してマッチングを試みます。これで名前の一致が見つかった場合、サービスから名前スコアが返されます。
4. **GlobalSentry Address Check** は、特定国内の住所に対してマッチングを試みます。これで住所の一致が見つかった場合、サービスから住所スコアが返されます。
5. **GlobalSentry ID Number Check** は、パスポート、国民背番号、社会保障番号 (SSN)、納税者番号などの識別番号に対してマッチングを試みます。識別番号の一致が見つかった場合、サービスから識別番号スコアが返されます。
6. 名前、住所、および識別番号の一致が見つからなかったトランザクションについては、トランザクション レコードが出力に書き出され、総合リスク水準スコアがゼロに設定されます。
7. 名前、住所、または識別番号の一致が見つかったトランザクションに対して、サービスは **GlobalSentry Other Data Check** サブフローを使用して **Global Sentry** データベースとの照合によるマッチングを試みます。
8. **GlobalSentry Other Data Check** は、出生地、出生日、国籍または市民権に対してマッチングを試みます。ここで一致が見つかった場合は、サービスから出生地スコア、出生日スコア、国籍スコア、または市民権スコアが返されます。
9. **GlobalSentry** は、各トランザクションに総合リスク水準スコアを設定します。このスコアは、0 ~ 16 の間の値を取り、**OverallRiskLevel** フィールドに返されます。リスク水準を計算するとき、**GlobalSentry** は入力レコードに含まれているデータの種別を考慮し、さらに **Global Sentry** データベースに一致するエントリがある場合には、それがどの入力に対応するかも考慮します。一般に、値が大きいほど、そのトランザクションに関係するリスクが高いことを示します。

## リソース URL

```
http://server:port/soap/GlobalSentry
```

## 例

### SOAP 要求:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:glob="http://www.pb.com/spectrum/services/GlobalSentry">
  <soapenv:Header/>
  <soapenv:Body>
    <glob:GlobalSentryRequest>
      <glob:options/>
      <glob:Input>
        <glob:Row>
          <glob:FirstName>Miguel</glob:FirstName>
        </glob:Row>
      </glob:Input>
    </glob:GlobalSentryRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <glob:LastName>Batista</glob:LastName>
      </glob:Row>
    </glob:Input>
  </glob:GlobalSentryRequest>
</soapenv:Body>
</soapenv:Envelope>

```

SOAP 応答は次のようになります。

注：この例では、空の応答要素を削除しています。最初の応答レコードのみを示しています。

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:GlobalSentryResponse
      xmlns:ns2="http://www.pb.com/spectrum/services/GlobalSentry">
      <ns2:Output>
        <ns2:Row>
          <ns2:OverallRiskLevel>10</ns2:OverallRiskLevel>

          <ns2:SanctionedCountryIdentified>No</ns2:SanctionedCountryIdentified>
          <ns2:Status>S</ns2:Status>
          <ns2:FirstName>Miguel</ns2:FirstName>
          <ns2:LastName>Batista</ns2:LastName>
          <ns2:PlaceOfBirth>San Sebastian (Guipuzcoa)
Spain</ns2:PlaceOfBirth>
          <ns2:EntryID>315</ns2:EntryID>

          <ns2:InputFilteredFirstName>Miguel</ns2:InputFilteredFirstName>

          <ns2:InputFilteredLastName>Batista</ns2:InputFilteredLastName>
          <ns2:InputFirstName>Miguel</ns2:InputFirstName>
          <ns2:InputLastName>Batista</ns2:InputLastName>
          <ns2:ListType>DFAT Consolidated List</ns2:ListType>
          <ns2:MatchKey1>MGL</ns2:MatchKey1>
          <ns2:MatchKey2>BTST</ns2:MatchKey2>
          <ns2:NameMatchIdentified>Yes</ns2:NameMatchIdentified>
          <ns2:NameProvided>Yes</ns2:NameProvided>
          <ns2:AddressProvided>No</ns2:AddressProvided>
          <ns2:IDNumberProvided>No</ns2:IDNumberProvided>

          <ns2:AddressMatchIdentified>No</ns2:AddressMatchIdentified>

          <ns2:IDNumberMatchIdentified>No</ns2:IDNumberMatchIdentified>
          <ns2:CitizenshipScore>0</ns2:CitizenshipScore>

          <ns2:CitizenshipMatchIdentified>No</ns2:CitizenshipMatchIdentified>
          <ns2:DOBScore>0</ns2:DOBScore>
          <ns2:DOBMatchIdentified>No</ns2:DOBMatchIdentified>
          <ns2:NationalityScore>0</ns2:NationalityScore>

          <ns2:NationalityMatchIdentified>No</ns2:NationalityMatchIdentified>

```

```

        <ns2:PlaceOfBirthScore>0</ns2:PlaceOfBirthScore>
<ns2:PlaceOfBirthMatchIdentified>No</ns2:PlaceOfBirthMatchIdentified>
    <ns2:CitizenshipProvided>No</ns2:CitizenshipProvided>
    <ns2:DOBProvided>No</ns2:DOBProvided>
    <ns2:NationalityProvided>No</ns2:NationalityProvided>
    <ns2:PlaceOfBirthProvided>No</ns2:PlaceOfBirthProvided>
    <ns2:WatchListFirstName>Miguel</ns2:WatchListFirstName>
    <ns2:WatchListLastName>ALBISU
IRIARTE</ns2:WatchListLastName>
    <ns2:NameScore>100</ns2:NameScore>
    <ns2:user_fields/>
    </ns2:Row>
  </ns2:Output>
</ns2:GlobalSentryResponse>
</soap:Body>
</soap:Envelope>

```

### 要求

入力データのパラメータ

表 73 : Global Sentry の入力フィールド

パラメータ	説明
Name	省略なしの完全な名前。 FirstName と LastName を使用しない場合は必須です。
FirstName	ファースト ネーム (姓名の名)、またはラスト ネーム (姓) 以外のすべての名前要素。 Name を使用しない場合は必須です。
LastName	ラスト ネーム (姓) のみ。 Name を使用しない場合は必須です。
AddressLine1	最初の住所行。 データありの場合は推奨項目です。

パラメータ	説明
AddressLine2	2 行目の住所行。 データありの場合は推奨項目です。
AddressLine3	3 行目の住所行。 データありの場合は推奨項目です。
Country	省略なしの完全な国名。 住所行を使用する場合は必須です。
IDNumber	SSN、パスポート、Visa などの識別番号。 データありの場合は推奨項目です。
PlaceOfBirth	任意の出生地データ。 データありの場合は推奨項目です。
DOB	誕生日 (Year, Month, Day の形式)。 データありの場合は推奨項目です。
Citizenship	省略なしの完全な国名。 データありの場合は推奨項目です。
Nationality	省略なしの完全な国名。 データありの場合は推奨項目です。



## 応答

表 74 : Global Sentry Service の出力

応答要素	説明
Status	マッチの成功または失敗。 <b>null</b> : 成功 <b>F</b> : 失敗
Status.Code	失敗の理由。
Status.Description	失敗の原因となった問題に関する説明。
名前	
InputName	元のデータ ソースからの入力名。
InputFilteredName	元のデータ ソースから肩書き、接尾語、および特殊文字を取り除いた入力名。
Name	データベースから返された名前。
InputFirstName	元のデータ ソースからの入力ファースト ネーム。
InputFilteredFirstName	元のデータ ソースから肩書き、接尾語、および特殊文字を取り除いた入力ファースト ネーム。
FirstName	データベースから返されたファースト ネーム。
InputLastName	元のデータ ソースからの入力ラスト ネーム。

応答要素	説明
InputFilteredLastName	元のデータ ソースから肩書き、接尾語、および特殊文字を取り除いた入力ラストネーム。
LastName	データベースから返されたラストネーム。
NameScore	名前一致スコア。0 ~ 100。
NameMatchIdentified	名前が一致したかどうかを示します。値は Yes または No です。
NameProvided	名前が入力データに含まれているかどうかを示します。値は Yes または No です。
<b>住所</b>	
InputAddressLine1	元のデータ ソースからの入力住所行。
AddressLine1	データベースから返された住所行。
InputAddressLine2	元のデータ ソースからの入力住所行。
AddressLine2	データベースから返された住所行。
InputAddressLine3	元のデータ ソースからの入力住所行。
AddressLine3	データベースから返された住所行。
AddressScore	住所一致スコア。0 ~ 100。
AddressMatchIdentified	住所が一致したかどうかを示します。値は Yes または No です。

応答要素	説明
AddressProvided	住所が入力データに含まれているかどうかを示します。値は Yes または No です。
InputCountry	元のデータ ソースからの入力国名。
Country	データベースからの国名。
<b>識別番号</b>	
InputIDNumber	元のデータ ソースからの入力識別番号。
IDNumber	データベースからの識別番号。
IDNumberScore	識別番号一致スコア。0 ～ 100。
IDNumberMatchIdentified	識別番号が一致したかどうかを示します。「はい」または「いいえ」。
IDNumberProvided	識別番号が入力データに含まれているかどうかを示します。値は Yes または No です。
<b>出生地</b>	
InputPlaceOfBirth	元のデータ ソースからの入力出生地。
PlaceOfBirth	データベースから返された出生地。
PlaceOfBirthScore	出生地一致スコア。0 ～ 100。
PlaceOfBirthMatchIdentified	出生地が一致したかどうかを示します。値は Yes または No です。

応答要素	説明
PlaceOfBirthProvided	出生地が入力データに含まれているかどうかを示します。値は Yes または No です。
<b>出生日</b>	
InputDOB	元のデータ ソースからの入力出生日。
DOB	データベースから返された出生日。
DOBscore	出生日一致スコア。0 ~ 100。
DOBMatchIdentified	出生日が一致したかどうかを示します。値は Yes または No です。
DOBProvided	出生日が入力データに含まれているかどうかを示します。値は Yes または No です。
<b>市民権</b>	
InputCitizenship	元のデータ ソースからの入力市民権。
Citizenship	データベースから返された市民権。
CitizenshipScore	市民権一致スコア。0 ~ 100。
CitizenshipMatchIdentified	市民権が一致したかどうかを示します。値は Yes または No です。
CitizenshipProvided	市民権が入力データに含まれているかどうかを示します。値は Yes または No です。

## 応答要素

## 説明

## 国籍

InputNationality

元のデータ ソースからの入力国籍。

Nationality

データベースから返された国籍。

NationalityScore

国籍一致スコア。0 ～ 100。

NationalityMatchIdentified

国籍が一致したかどうかを示します。値は Yes または No です。

NationalityProvided

国籍が入力データに含まれているかどうかを示します。値は Yes または No です。

## 政府リスト情報

EntryID

名前、団体、船舶、住所、識別番号、出生地、出生日、市民権、または国籍を示すエントリ ID。これは各政府機関から提供されます。

ListType

政府機関から提供されるリストの名前。SDN、EU、イングランド銀行、カナダ金融機関監督庁。

## リスク解析

OverAllRiskLevel

一致ごとのリスクソース。0 ～ 16。詳細については、「[リスク解析スコアについて \(430ページ\)](#)」を参照してください。

SanctionedCountryIdentified

認可国が一致すると認識されたかどうかを示します。値は Yes または No です。

## リスク解析スコアについて

これらの各入力に対しては、入力の有無と Global Sentry データベース内に一致するレコードがあるかどうかに応じて、リスク解析処理が一定のポイント値を割り当てます。リスク解析スコアは、これらのポイント値の合計です。ポイントがどのように割り当てられるかを次の表に示します。

**表 75: リスク解析のスコアリング方法**

入力	入力データなし	一致あり	一致なし
名前	0	4	0
住所	1	2	0
ID	1	2	0
誕生日	1	2	0
出生地	1	2	0
市民権	1	2	0
国籍	1	2	0

一般に、データベースと一致した各入力には 2 ポイントが割り当てられます。ただし、名前は例外です。名前の一致は 4 ポイントとカウントされます。名前スコアの加重が高いのは、OFAC などの情報源から得られる指針に従っているからで、そこでは名前の一致はその他の種類の一致よりもより重要であるとされています。

一方、入力があっても、データベースのどのエントリとも一致しない場合、割り当てられるポイントは 0 で、総合リスク水準には影響しません。これは、名前の一致があっても、その他の相当量のデータがデータベース内のエントリと一致しなければ、その一致を特定リストとの照合による "ヒット" とは見なすべきでないという指針に従うものです。

データの無い入力に 1 ポイントが設定されています。これには、1 つ以上の入力がデータベースと一致する一方で、マッチングに使用できる入力が一部欠けているようなトランザクションを、

よりリスクが高いトランザクションであると見なす効果があります。このタイプのトランザクションについては、データが欠けているが故に真のリスク水準を正確に計算できません。OFACなどの機関から得られる指針によれば、こうしたケースではトランザクションに関するリスクのより正確な評価を与えるために、欠けているデータの入手に努力することになっています。

スコアが高ければそれだけトランザクションのリスクも大きくなりますが、リスク水準だけで必ずしも適切なアクションを決定できるわけではありません。これはスコアが同じでも、一致あり、一致なし、データなしの入力について、さまざまな組み合わせが考えられるからです。禁止措置の妥当性を判断する補足的な情報を提供するために、Global Sentry サービスはマッチングで使われる7つの入力のそれぞれについて、さらに2つの指標を返します。これらは入力データの有無と、データベースの一致の有無を示します。これで、リスク範囲の中程にあるトランザクションに対して追加的な解析を行うことにより、ブラックリストの管理当局にトランザクションを報告する、正確なリスク評価のために追加的な入力データが必要とのマークをトランザクションに付ける、トランザクションを承認する、または、その他の何らかのアクションを行うなど、どの対処が妥当かを判断できます。

### Global Sentry サービスのカスタマイズ

Global Sentry は、Enterprise Designer で部分的に修正できる5つのデータフロー テンプレートを展開しています。個々のデータフローは、Spectrum™ Technology Platform、Universal Name、Data Normalization、Advanced Matching の各モジュールからインストールされた各種のコンポーネントによって構成されます。

以下のデータフローが使用可能です。

- 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

## Information Extraction モジュール

### InformationExtractor

InformationExtractor は、構造化されていないデータの文字列 (プレーンテキスト) から、名前や住所などのエンティティを抽出します。

入力のタイプによって精度が異なるため、選択されたタイプに対して一部のエンティティは返されない可能性があります。Information Extractor は自然言語処理を使用するため、ニュース記事やブログからの文法的に正しい文を含む文字列は、名前や日付の単純なリストよりも、高い精度で名前が返される可能性があります。

### リソース URL

```
http://server:port/soap/InformationExtractor
```

### 例

以下は、SOAP リクエストです。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:inf="http://www.pb.com/spectrum/services/InformationExtractor">
  <soapenv:Header/>
  <soapenv:Body>
    <inf:InformationExtractorRequest>
      <inf:options>
        <inf:EntityList>Person</inf:EntityList>
      </inf:options>
      <inf:input_port>
        <inf:PlainText>
          <inf:PlainText>My name is Arthur Pitney</inf:PlainText>
        </inf:PlainText>
        <inf:PlainText>
          <inf:PlainText>My name is Walter Bowes</inf:PlainText>
        </inf:PlainText>
      </inf:input_port>
    </inf:InformationExtractorRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:InformationExtractorResponse
```



```

xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/InformationExtractor">
  <ns3:output_port>
    <ns3:Result>
      <ns3:Entity>
        <ns3:Entity>
          <ns3:Text>Arthur Pitney</ns3:Text>
          <ns3:Type>Person</ns3:Type>
        </ns3:Entity>
      </ns3:Entity>
      <ns3:user_fields/>
    </ns3:Result>
    <ns3:Result>
      <ns3:Entity>
        <ns3:Entity>
          <ns3:Text>Walter Bowes</ns3:Text>
          <ns3:Type>Person</ns3:Type>
        </ns3:Entity>
      </ns3:Entity>
      <ns3:user_fields/>
    </ns3:Result>
  </ns3:output_port>
</ns3:InformationExtractorResponse>
</soap:Body>
</soap:Envelope>

```

### 要求

#### 入力データのパラメータ

InformationExtractor は、構造化されていないデータを入力として受け取ります。

### 表 76 : 入力フォーマット

パラメータ	説明
-------	----

PlainText	情報の抽出元となるデータを表す構造化されていない文字列です。
-----------	--------------------------------

### オプション

InformationExtractor ステージを使用すると、出力データのためのエンティティを選択できます。このステージに取り込まれたエンティティ タイプについては、属性が自動的に割り当てられます。ただし、クイック追加機能を使用して次の 15 の属性の任意のものまたはすべてを選択できます。

パラメータ	説明
CategoryName	テキスト分類に使用するモデルを指定します。
CategoryCount	出力するカテゴリのマッチングレベル数(最も近い一致、最も近いものと2番目に近いもの、など)を指定します。
EntityList	<p>構造化されていない文字列から抽出するデータのタイプを指定します。</p> <p>1つまたは複数指定します。各エンティティタイプの区切りにはカンマを使用します。</p> <ul style="list-style-type: none"><li>Address</li><li>CreditCard</li><li>Date</li><li>Email</li><li>HashTag</li><li>ISBN</li><li>Location</li><li>Mention</li><li>Organization</li><li>Person</li><li>Phone</li><li>ProperNouns</li><li>SSN</li><li>WebAddress</li><li>ZipCode</li></ul>
OutputEntityCount	<p>特定のエンティティが現れた回数を出力で返すかどうかを指定します。</p> <ul style="list-style-type: none"><li><b>true</b> 構造化されていない文字列内に見つかったエンティティの数を返します。</li><li><b>false</b> 構造化されていない文字列内に見つかったエンティティの数を返しません。</li></ul>

## 応答

InformationExtractor からの出力は、入力文字列内に見つかったエンティティの一覧です。例えば、"Person" というエンティティ タイプを選択した場合、出力は入力文字列内に見つかった名前の一覧になります。同様に、"Date" というエンティティ タイプを選択した場合は、入力文字列内に見つかった日付の一覧が出力になります。それぞれのエンティティ (名前、住所、日付など) は、たとえ入力文字列内に複数回現れていても、一度しか返されません。

応答要素	説明
Text	文字列から抽出されたテキストです。
Type	抽出されたテキストのエンティティ タイプです。次のいずれかです。 <b>Address</b> <b>CreditCard</b> <b>Date</b> <b>Email</b> <b>HashTag</b> <b>ISBN</b> <b>Location</b> <b>Mention</b> <b>Organization</b> <b>Person</b> <b>Phone</b> <b>ProperNouns</b> <b>SSN</b> <b>WebAddress</b> <b>ZipCode</b>
Count	回数を返すオプションが有効になっている場合は、このフィールドに、その特定のエンティティが入力に現れた回数が含まれます。例えば、名前エンティティを返すことを選択していて、入力テキストに "John" という名前のインスタンスが 5 つ含まれている場合、出力には "John" が 1 回だけ含まれ、エンティティ タイプは "Name" に、出力回数は "5" になります。
Category	カテゴリーを使用した場合、入力ファイル内の各レコードに対する予測カテゴリーです。

応答要素	説明
Rank	カテゴリザを使用した場合は、最多カウントから最少カウントまでのカテゴリのランクです。

## Spatial モジュール

### ドキュメントの場所

Location Intelligence モジュールは、任意の位置、対象領域、または対象地点やその他のビジネスデータ間の関係を決定し、そうした関係をマップ上に視覚的に示すことができる空間サービスを提供します。これらのサービスには次のものが含まれます。

- Geometry
- Feature
- Mapping
- MapTiling
- Named Resource
- Web Feature Service
- Web Map Service

Location Intelligence モジュール サービスの詳細については、『*Spectrum Spatial ガイド*』([support.pb.com](http://support.pb.com) にあります)を参照してください。

### GetTravelBoundary (非推奨)

**重要:** このステージは 12.2 リリースで非推奨になっています。新しいデータフローを作成するときは、代わりに **到達可能エリア** ステージを使用してください。

GetTravelBoundary は、ロケーションから一定の時間または距離で、車両または徒歩で到達可能なエリアの境界を示します。この機能では、時間到達圏または距離到達圏の計算に基づくポリゴンを取得します。時間到達圏とは、始点から一定時間内に道路を通過して到達できるエリアを表すポリゴンまたはポイントの集まりです。距離到達圏とは、始点から一定距離の移動で到達できるエリアを表すポリゴンまたはポイントの集まりです。Get Travel Boundary 操作 (到達圏定義とも呼ばれます) は、始点、単位 (距離または時間)、1つ以上のコスト、およびそれらに関連付けられたタグを入力として受け取り、生成された到達圏境界を返します。コストとは、到達圏計算に使用する時間または距離の量を意味します。タグは、コストを一意に指す文字列で、対応する結果とのマッチングに使用されます。複数のコストを“;”区切り文字列として提供することで、コストを入力として指定できます。

GetTravelBoundary は、Spatial モジュールに含まれています。

注：GetTravelBoundary は、Web サービスとしてのみ利用可能です。GetTravelBoundary を、Java、C++、C、.NET、または COM API を介して利用することはできません。

### リソース URL

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

### 例

ケース 1、単一のコスト:

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

応答は次のようになります。

注：この例では、短縮のため一部のポイントが削除されています。

```
<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>

```

## ケース 2、複数のコスト:

```

<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:IsoRouteRequest>
      </get:input_port>
    </get:GetTravelBoundaryRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

応答は次のようになります。

注：この例では、短縮のため一部のポイントが削除されています。

```

<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>
```

## 要求

### 入力データのパラメータ

GetTravelBoundary は、コスト、コスト単位、ポイント緯度、およびポイント経度を入力として受け取ります。以下の表に、入力のフォーマットとレイアウトに関する情報を示します。

表 77 : GetTravelBoundary 入力データ

パラメータ	書式	説明
Latitude	文字列	ポイントの緯度。CoordinateFormat オプションで選択されたフォーマットを使って緯度を指定します。
Longitude	文字列	ポイントの経度。ConditionalFormat オプションで選択されたフォーマットを使って経度を指定します。
TravelBoundaryCost	文字列	<p>(オプション) TravelBoundaryCostUnits フィールドまたは DefaultTravelBoundaryCostUnits オプションで指定された単位で表された距離または時間のコスト。例えば、指定された単位がマイルのときにこのフィールドに 10 を指定すると、コストは 10 マイルになります。</p> <p>このフィールドを使って、デフォルト経路境界コストをレコード単位で上書きできます。</p> <p>また、複数のコストを指定することも可能で、値を ";" で区切られた文字列として指定します。指定されたすべてのコストに対して個別に Iso Route Response が返されます。複数のコストを指定した場合、応答にはコストとその応答に関連付けられている costUnits が含まれます。</p>

パラメータ	書式	説明
TravelBoundaryCostUnits	文字列	<p>(オプション)経路境界の計算に使用される計測単位のタイプ。次のいずれかです。</p> <ul style="list-style-type: none"> <li>• Feet</li> <li>• Hours</li> <li>• Kilometers</li> <li>• Meters</li> <li>• Miles</li> <li>• Minutes</li> <li>• Seconds</li> <li>• Yards</li> </ul> <p>このフィールドを使って、デフォルト経路境界コスト単位をレコード単位で上書きできます。</p>

### オプションのパラメータ

入力

表 78 : GetTravelBoundary の入力オプション

パラメータ	説明
DataSetResourceName	<p>検索プロセスで使用するデータが格納されたデータベースの名前。Management Console の [リソース] セクションで定義した有効なルーティング データベース リソース名を使用します。詳細については、『<i>Spectrum™ Technology Platform Spatial</i> ガイド』を参照してください。</p>
CoordinateSystem	<p>入力される座標の座標系。</p> <p>EPSG コードの詳細については、<a href="http://www.spatialreference.org">www.spatialreference.org</a> を参照してください。EPSG に対してサポートされるコードスペースのリストを取得するには、SOAP リクエスト List Supported CoordSys by Code Space を Geometry サービス デモ ページ <code>http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html</code> から送信します。</p>

## パラメータ

## 説明

---

**CoordinateFormat**

入力座標の緯度/経度のフォーマットを指定します。

注：このオプションは、緯度/経度座標系を指定した場合のみに使用します。座標系が緯度/経度座標系ではない場合は、座標フォーマットを [小数] に設定します。

次のいずれかです。

<b>Decimal</b>	(90.000000, 180.000000)
<b>DecimalAssumed</b>	(90000000, 180000000)。こちらがデフォルトです。
<b>DegreesMinutesSeconds</b>	(90 00 00N, 180 00 00W)
<b>PreZero</b>	(090000000N, 180000000W)
<b>PreZeroDecimal</b>	(090.000000N, 180.000000W)

---

**DefaultTravelBoundaryCost**コスト単位の値。任意の **double** 型の値 (小数を含む) を指定できます。デフォルトは **10** です。

---

**DefaultTravelBoundaryCostUnits**

経路境界の計算に使う測定単位。次のいずれかです。

- Feet
  - Hours
  - Kilometers
  - Meters
  - Miles
  - Minutes
  - Seconds
  - Milliseconds
  - Yards
-

## パラメータ

## 説明

## HistoricTrafficTimeBucket

ルーティング計算において、トラフィック速度履歴を使用するかどうかを指定します。これらの速度は、1日のうちの異なるタイム バケットに基づきます。この機能を使用するには、トラフィック速度履歴がデータに含まれている必要があります。各国/地域のデータに対するバケット定義は同じです。バケット値に対する速度は異なる場合があります。次のオプションがあります。

- None** デフォルト値です。計算において、トラフィックデータ履歴は使用しません。代わりに速度の平均値を使用します。
- AMPeak** 午前のピーク速度でルートを計算します。午前ピークのタイム バケットは、1日のうちの 07:00 ~ 10:00 です。
- PMPeak** 午後のピーク速度でルートを計算します。午後ピークのタイム バケットは、1日のうちの 16:00 ~ 19:00 です。
- OffPeak** オフピーク (日中) 速度でルートを計算します。オフピークのタイム バケットは、1日のうちの 10:00 ~ 16:00 です。
- Night** 夜間速度でルートを計算します。夜間のタイム バケットは、1日のうちの 22:00 ~ 04:00 です。

## 出力

表 79 : GetTravelBoundary の出力オプション

## パラメータ

## 説明

## ResultType

返す結果のタイプを指定します。次のいずれかです。

- AccessibleNodes** isoChrone 全体を返します。
- Geometry** isoChrone 計算に到達しうる、道路沿いのすべてのポイントを返します。
- StartNodes** 入力される住所で指定される場所を返します。

パラメータ	説明
SimplificationFactor	元のポイントをどの割合で返す必要があるか(結果のポリゴンの基になるポイントの割合)を指定します。
BandingStyle	結果に使用する境界線表示形式を指定します。境界線表示形式は、複数のコストに基づいて表示できる複数の isoChrono または境界線のタイプです。 <b>Donut</b> 境界は、次に小さな境界をくり抜く方法で確定されます。 <b>Encompassing</b> 境界は、他のすべての境界から独立して確定されます。
ReturnHoles	境界内の到達不可能領域を返すかどうかを指定します。これは、道順に従って適切な時間または距離で到達できない、より大きな境界内に存在する領域を指します。 <b>Y</b> 境界内の到達不可能領域を返します。 <b>N</b> 境界内の到達不可能領域を返しません。こちらがデフォルトです。
ReturnsIslands	境界外の到達可能領域を返すかどうかを指定します。これは、適切な時間または距離で到達できない、主要境界の外にある小さな領域です。 <b>Y</b> 境界外の到達可能領域を返します。 <b>N</b> 境界外の到達可能領域を返しません。こちらがデフォルトです。

## 経路

経路オプションでは、道路以外を移動する際の移動速度や、経路境界を計算する際に幹線道路のみを走行するかどうかなどを指定します。多くの経路オプションは、周辺移動速度に関連するものです。

表 80 : GetTravelBoundary の経路オプション

パラメータ	説明
MaximumOffRoadDistance	道路以外を移動する距離の最大値を指定します。道路以外とは、私道や進入路などのことです。例えば、道路外距離の最大値を 1 マイルと指定すると、道路から 1 マイル以上離れた地点は、到達圏境界に含まれません。道路外距離の最大値として 0 を指定すると、道路から少しでも離れた地点は到達圏境界に含まれません。道路以外を移動する際の速度は、周辺移動速度オプションによって指定します。
Units	返されるデータの単位を指定します。次のいずれかです。 <ul style="list-style-type: none"><li>• キロメートル (デフォルト)</li><li>• メートル</li><li>• マイル</li></ul>

パラメータ

説明

---

MajorRoads

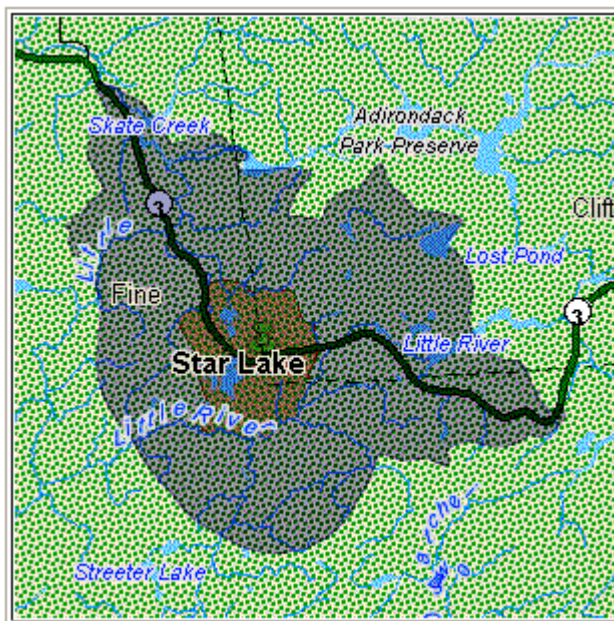


## パラメータ

## 説明

計算時に、すべての道路を含めるか、幹線道路のみを含めるかを指定します。デフォルトでは、幹線道路が True に設定され、Get Travel Boundary が計算されます。これによりパフォーマンスが向上しますが、精度は低下する可能性があります。

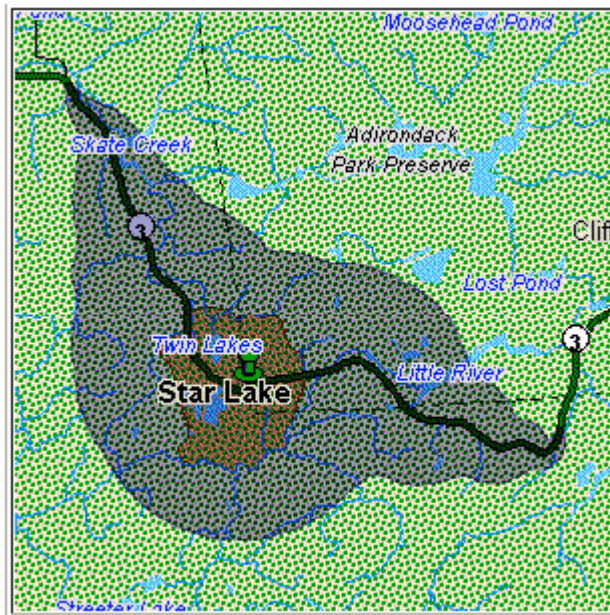
次のマップは、すべての道路を走行する場合の経路境界を表します。



次のマップは、幹線道路のみを走行する場合の経路境界を表します。

## パラメータ

## 説明



次のいずれかです。

- Y** 計算時に、幹線道路のみを含めます。デフォルト
- N** 計算時に、すべての道路を含めます。

## DefaultAmbientSpeed

経路境界を検索するために使用する、道路以外を移動する際の速度を指定します。道路以外とは、私道や進入路などのことです。

このオプションは、DefaultCostUnits オプション、または TravelBoundaryCostUnits フィールドで時間値を指定した場合のみ有効です。デフォルト値は 15 です。速度の単位は AmbientSpeedUnit オプションで指定します。

到達圏境界計算において道路外の移動の処理を制御するには、道路外を移動する際の速度 (周辺移動速度) を指定する必要があります。周辺移動速度は、到達圏境界を表すポリゴンの大きさや形状に影響を与える可能性があります。一般的には、周辺移動速度が速いほど、ポリゴンは大きくなります。例えば、あと 5 分の移動が可能で、周辺移動速度が 15 マイル/時の場合は、到達可能地点まで 1.25 マイルの距離があることとなります。周辺移動速度を 10 マイル/時に落とすと、到達可能地点までの距離は 0.83 マイルになります。

MaximumOffRoadDistance オプションを使用することにより、道路外を移動する距離を制限できることに注意してください。

注：歩行者の経路境界を計算する場合は、デフォルト周辺移動速度を 3 MPH (5 KPH) に変更することを推奨します。

## パラメータ

## 説明

---

AmbientSpeedUnit	DefaultAmbientSpeed オプションで指定する値に使用する単位を指定します。
<b>KPH</b>	キロメートル/時
<b>MPH</b>	マイル/時デフォルト
<b>MTPS</b>	メートル/秒
<b>MTPM</b>	メートル/分

---

パラメータ

説明

---

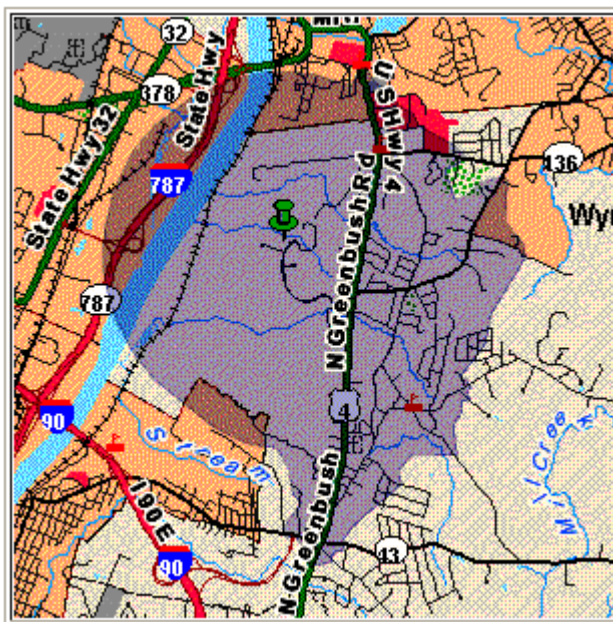
AmbientSpeed.RoadType.<Type>

## パラメータ

## 説明

道路の種類ごとに、道路外を移動する際の周辺移動速度を指定します。道路のタイプに対する周辺移動速度を指定しない場合は、DefaultAmbientSpeed オプションで指定されたデフォルト周辺移動速度が適用されます。

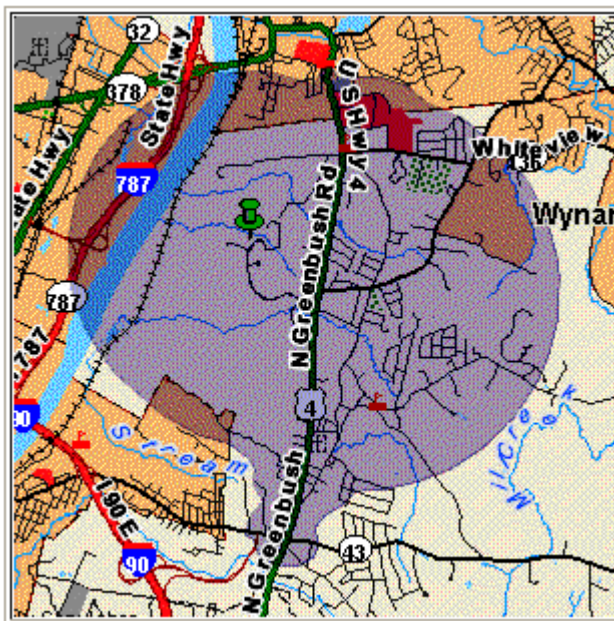
次のマップは、周辺移動速度のオーバーライド値を指定しない場合の経路境界の例を示しています。



比較のために、周辺移動速度のオーバーライド値を指定した場合の、上と同じ経路境界を次のマップに示します。

パラメータ

説明



<Type> には、次の種類があります。



## パラメータ

## 説明

- 
- AccessWay
  - Backroad
  - Connector
  - Ferry
  - Footpath
  - LimitedAccessDenseUrban
  - LimitedAccessRural
  - LimitedAccessSuburban
  - 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
-

## 応答

Get Travel Boundary は、次のフィールドを返します。

表 81 : GetTravelBoundary の出力

応答要素	書式	説明
Status	文字列	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	文字列	失敗の原因 (ある場合)。次のいずれかです。 <ul style="list-style-type: none"> <li>• InsufficientInputData (緯度/経度の欠落)</li> <li>• MalformedInputData (誤った入力フォーマット)</li> <li>• InputOutOfRange (範囲外の入力)</li> <li>• EngineError (エンジン由来のエラー)</li> </ul>
Status.Description	文字列	Status.Code で示される失敗に関する説明。

## GetTravelCostMatrix (非推奨)

GetTravelCostMatrix は、複数の始点と終点の間の移動時間と距離を計算します。

GetTravelCostMatrix を使用して、複数の始点から複数の終点の間の最短または最速ルートを検索し、個々のルートの合計時間および距離 (ルート コスト) を算出できます。例えば、4 つの始点 (S1 ~ S4) と 4 つの終点 (E1 ~ E4) を入力した場合、次の図に示すように合計 16 件のルートが返されます。



マトリクスルーティングは主に、消防署や警察署といった特定のサービスの応答時間および範囲を特定するために使用します。最初の通報から 20 分以内に事故または事件に対応できる人員を 1 人以上配備するといったサービス レベル アグリーメント (SLA) の、正確な保証範囲を提示するために、このような計算が必要になる場合があります。この計算を分析ツールとして使用するこ



とにより、救急車、警察官、または消防隊員が、合理的な時間以内に不動産/人物まで駆けつけられる確率に基づいて、保険対象の不動産および人物のリスクを判定できます。

注：Get Travel Cost Matrix は、SOAP Web サービスとしてのみ利用可能です。Get Travel Cost Matrix を、REST を介して利用することはできません。Java、C++、C、.NET、または COM API を介して利用することもできません。

GetTravelCostMatrix は、Enterprise Routing モジュールに含まれています。

## リソース URL

```
http://server:port/soap/GetTravelCostMatrix
```

## 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:get="http://www.pb.com/spectrum/services/GetTravelCostMatrix"

  xmlns:spec="http://spectrum.pb.com/"
  xmlns:get1="http://www.g1.com/services/GetTravelCostMatrix"
  xmlns:typ="http://www.g1.com/services/erm/types">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetTravelCostMatrixRequest>
      <get:input_port>
        <get:RouteMatrixRequest>
          <get:StartPoints>
            <get:StartPoint>
              <get:Latitude>33.751748</get:Latitude>

              <get:Longitude>-84.364014</get:Longitude>
            </get:StartPoint>
            <get:StartPoint>
              <get:Latitude>33.870416</get:Latitude>
              <get:Longitude>-78.62915</get:Longitude>
            </get:StartPoint>
            <get:StartPoint>
              <get:Latitude>35.025498</get:Latitude>
              <get:Longitude>-80.864868</get:Longitude>
            </get:StartPoint>
          </get:StartPoints>
          <get:EndPoints>
            <get:EndPoint>
              <get:Latitude>33.664925</get:Latitude>
              <get:Longitude>-80.90332</get:Longitude>
            </get:EndPoint>
            <get:EndPoint>
```

```

        <get:Latitude>34.40691</get:Latitude>
        <get:Longitude>-80.062866</get:Longitude>
    </get:EndPoint>
    <get:EndPoint>
        <get:Latitude>34.921971</get:Latitude>
        <get:Longitude>-81.013184</get:Longitude>
    </get:EndPoint>
    </get:Endpoints>
</get:RouteMatrixRequest>
</get:input_port>
</get:GetTravelCostMatrixRequest>
</soapenv:Body>
</soapenv:Envelope>

```

応答は次のようになります。

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns5:GetTravelCostMatrixResponse
xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.g1.com/services/erm/types"
xmlns:ns4="http://www.g1.com/services/GetTravelCostMatrix"

xmlns:ns5="http://www.pb.com/spectrum/services/GetTravelCostMatrix">
      <ns5:output_port>
        <ns5:RouteMatrixResponse>
          <ns5:TimeUnits>Minutes</ns5:TimeUnits>
          <ns5:DistanceUnits>Miles</ns5:DistanceUnits>
          <ns5:RouteCosts>
            <ns5:RouteCost>
              <ns5:StartPointRef>1</ns5:StartPointRef>
              <ns5:EndPointRef>1</ns5:EndPointRef>
              <ns5:Time>215.82</ns5:Time>
              <ns5:Distance>218.441</ns5:Distance>
            </ns5:RouteCost>
            <ns5:RouteCost>
              <ns5:StartPointRef>2</ns5:StartPointRef>
              <ns5:EndPointRef>2</ns5:EndPointRef>
              <ns5:Time>124.82</ns5:Time>
              <ns5:Distance>103.437</ns5:Distance>
            </ns5:RouteCost>
            <ns5:RouteCost>
              <ns5:StartPointRef>3</ns5:StartPointRef>
              <ns5:EndPointRef>3</ns5:EndPointRef>
              <ns5:Time>22.53</ns5:Time>
              <ns5:Distance>15.005</ns5:Distance>
            </ns5:RouteCost>
          </ns5:RouteCosts>
          <ns5:user_fields/>
        </ns5:RouteMatrixResponse>
      </ns5:output_port>
    </ns5:GetTravelCostMatrixResponse>
  </soap:Body>
</soap:Envelope>

```

```
</soap:Body>
</soap:Envelope>
```

### 要求

#### 入力データのパラメータ

Get Travel Cost Matrix への入力は、緯度/経度の座標で表された一連の始点と終点です。また、ユーザ定義フィールドを含めることもできます。

入力する始点と終点の順番によって、応答の順番が決まります。例えば、要求で 2 つの始点と 2 つの終点を指定した場合、応答に含まれる順番は、S1 から E1、S1 から E2、S2 から E1、S2 から E2 となります (S は始点、E は終点を表します)。

注： Get Travel Cost Matrix は、Web サービスとしてのみ利用可能です。Java、C++、C、.NET、または COM API を介して Get Travel Cost Matrix を利用することはできません。

表 82 : Get Travel Cost Matrix 入力

パラメータ	説明
Latitude	始点または終点の緯度。CoordinateFormat オプションで選択されたフォーマットを使って緯度を指定します。
Longitude	始点または終点の経度。CoordinateFormat オプションで選択されたフォーマットを使って経度を指定します。
ID	ポイントに割り当てる識別子。ポイントを表す英数字の ID を指定します。この ID は、出力の [StartPointID] フィールドまたは [EndPointID] フィールドに対応します。
TollRoad	この機能では、ルートに有料道路を含めるかどうかを指定できます。これは Boolean 型のパラメータです。デフォルト値は False です。TollRoad の値を True に設定した場合、レスポンスには有料道路が一切含まれていないルートが格納されます。TollRoad の値が False に設定されている場合、ルートには有料道路が含まれます。

パラメータ	説明
-------	----

---

<b>MatrixTransientUpdate</b>
------------------------------

一時更新の更新タイプを含むスキーマです。一時更新は、特定の要求にのみ適用される更新です。一時更新は永続更新に似ていますが、一時更新は特定の要求のみに対応し、永続更新はすべての要求に対応しています。ポイントの速度、セグメント ID、または道路クラスを設定できるだけでなく、セグメント (セグメント ID で指定) の道路クラスを更新できます。

一時更新のオプションと例

は、[../GetTravelCostMatrix/GetTravelCostMatrix\\_TransientOptions.dita](#)を参照してください。

---

### オプションのパラメータ

#### 経路

この優先設定セットを使用して、道路タイプごとに望ましさを設定することができます。例えば、サーバーがすべての幹線道路タイプを回避するよう要求することができます。

表 83 : 経路の優先設定オプション

パラメータ	説明
-------	----

---

RoadType_<type>	
-----------------	--

## パラメータ

## 説明

---

ルート決定時にさまざまな道路に与える優先順位を指定します。

- access way
- back road
- connector
- ferry
- footpath
- limited access dense urban
- limited access rural
- limited access suburban
- limited access urban
- local road dense urban
- local road rural
- local road suburban
- local road urban
- major local road dense urban
- major local road rural
- major local road suburban
- major local road urban
- major road dense urban
- major road rural
- major road suburban
- major road urban
- minor local road dense Urban
- minor local road rural
- minor local road suburban
- minor local road urban
- normal road dense urban
- normal road rural
- normal road rural
- normal road urban
- primary highway dense urban
- primary highway rural
- primary highway suburban
- primary highway urban
- ramp dense urban
- ramp limited access
- ramp major road
- ramp primary highway
- ramp rural
- ramp secondary highway
- ramp urban
- ramp suburban
- secondary highway dense urban
- secondary highway rural

## パラメータ

## 説明

- secondary highway suburban
- secondary highway urban

道路タイプごとに、次のいずれかを指定できます。

**Avoid** この道路タイプを可能な限りルートで使用しません。

注：特定の道路タイプを道順から常に除外できるとは限りません。状況によっては、回避対象の道路タイプの代わりになるものが非常に貧弱なために回避対象の道路タイプを使うルートが選ばれることがあります。また、始点または終点がセグメントの横にあって、そのセグメントの道路タイプが回避されている場合も、そのセグメントが使われます。

**High** この道路タイプを他の道路タイプよりも優先します。

**Low** 他の道路タイプをこの道路タイプよりも優先します。

**Medium** この道路タイプに他の道路タイプと同じ優先設定を与えます。道路タイプに優先設定を指定しないと、デフォルトとして "中" が使われます。

パラメータ

説明

---

MajorRoads

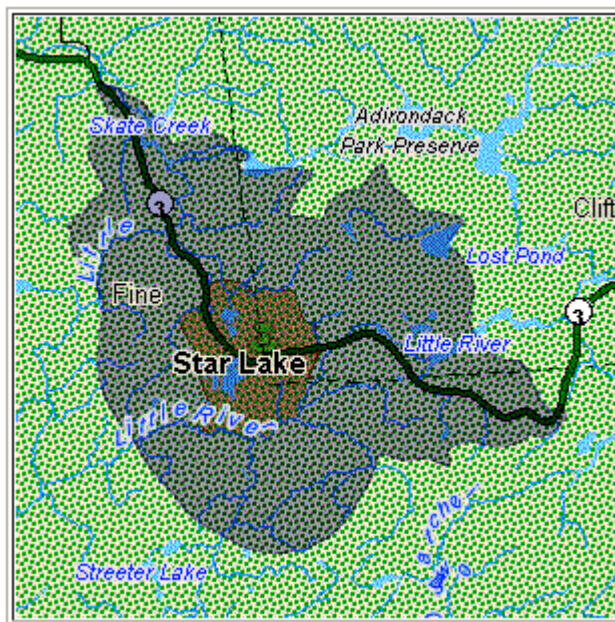


## パラメータ

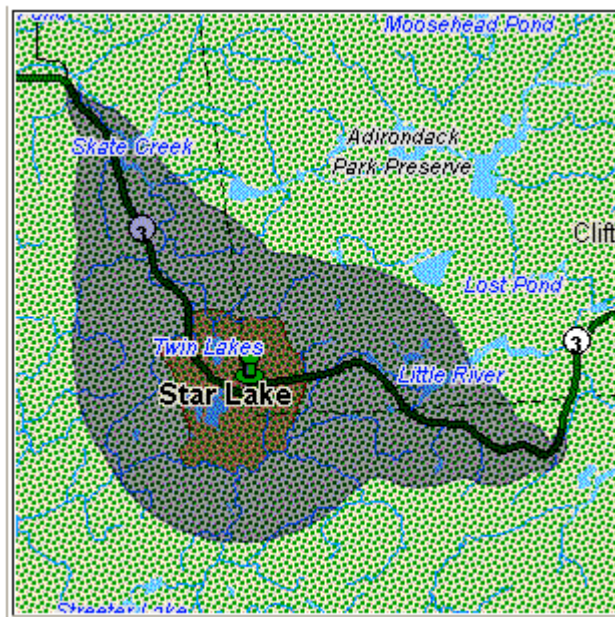
## 説明

計算時に、すべての道路を含めるか、幹線道路のみを含めるかを指定します。幹線道路のみを含めるように指定した場合は、パフォーマンスは向上しますが精度は低くなる場合があります。

次のマップは、すべての道路を走行する場合の経路境界を表します。



次のマップは、幹線道路のみを走行する場合の経路境界を表します。



## パラメータ

## 説明

---

次のいずれかです。

- Y** 計算時に、幹線道路のみを含めます。デフォルト
  - N** 計算時に、すべての道路を含めます。
-

パラメータ

説明

---

除外

有料道路

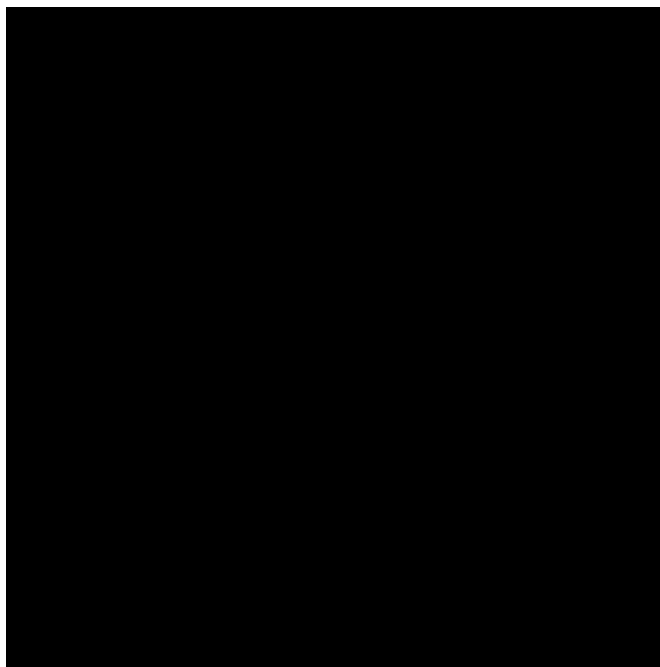
## パラメータ

## 説明

ルートに有料道路を含めるかどうかを指定できます。GetTravelCostMatrix GetTravelDirections ステージには、「有料道路の回避」機能が含まれています。UIに [有料道路] というラベルの付いたチェックボックスがあります。このチェックボックスをオンにすると、有料道路を回避できます。入力値で "TollRoad" としてこのパラメータを追加またはエクスポートすることもできます。入力値には Boolean 値を含むことができます。デフォルト値は False です。

**例:**

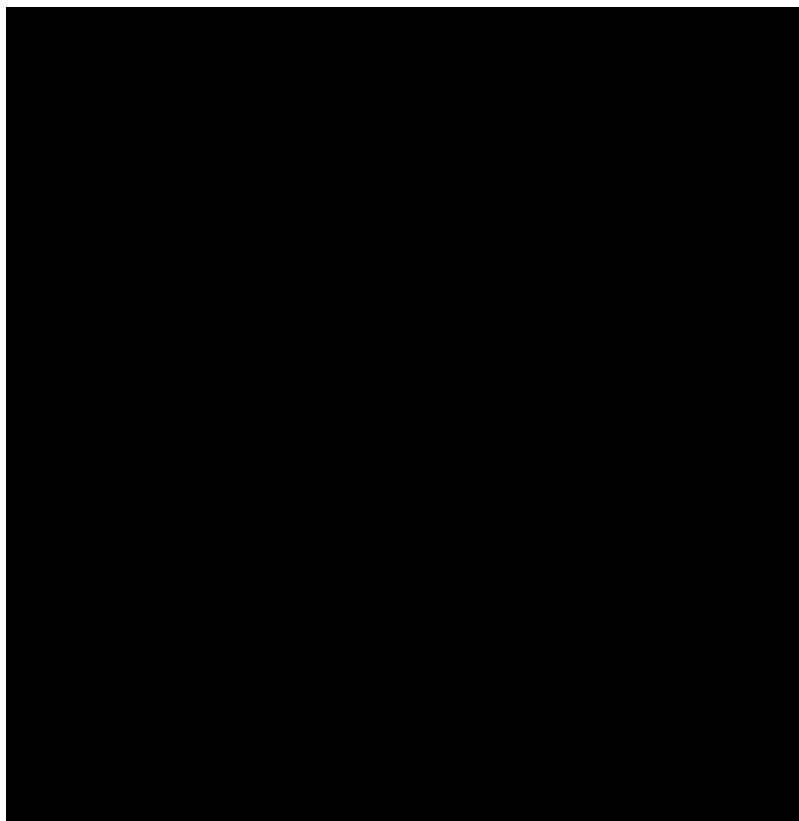
以下のルートには、有料道路情報が含まれています。以下の図を参照してください。



同じポイントに対して、[有料道路] チェック ボックスをオンにするか、または "TollRoad" パラメータの値を True に設定した場合、レスポンスには有料道路が一切含まれていないルートが格納されます。次のレスポンスを参照してください。

パラメータ	説明
-------	----

---



---

ルーティング

表 84 : ルーティング オプション

Parameter	説明
DataSetResourceName	検索プロセスで使用するデータが格納されたデータベースの名前。 Management Console の [リソース] セクションで定義した有効なルーティン グ データベース リソース名を使用します。詳細については、『 <i>Spectrum™ Technology Platform Spatial</i> ガイド』を参照してください。

---

Parameter	説明
OptimizeBy	<p>最短距離または最短時間のどちらを検索するかを指定します。</p> <p>次のいずれかです。</p> <p><b>Time</b> 移動時間が最短となるように最適化します。デフォルト</p> <p><b>Distance</b> 移動距離が最短となるように最適化します。</p>
ReturnOptimalRoutesOnly	<p>始点/終点の各組み合わせに対し、最適化ルートのみを返すかどうかを指定します。最適化ルートとは、OptimizeBy オプションの選択によって、最速ルートまたは最短ルートのいずれかになります。</p> <p>例えば、1つの始点 (S1) と 2つの終点 (E1 および E2) があり、S1 から E2 への移動時間の方が S1 から E1 への移動時間よりも短く、かつ、最適なルートのみを返すと指定した場合は、S1 から E2 へのルート コスト情報のみが返されます。</p> <p><b>Y</b> 最適なルートのみを返します。デフォルト</p> <p><b>N</b> 最適なルートのみでなく、すべてのルートを返します。</p>
RouteCostMatrixFormat	<p>ルート コスト マトリクスのデータ フォーマットを指定します。次のいずれかです。</p> <p><b>Hierarchy</b> ルート コスト マトリクスは、Management Console または Enterprise Designer で表示可能なリストとして返されます。これらのツールでデータを表示する場合、または、テーブル形式でデータを処理する場合は、このオプションを選択します。デフォルト</p> <p><b>Object</b> ルート コスト マトリクスは、直接操作することのできないデータ オブジェクトとして返されます。出力をデータフローや Web サービス呼び出しの他のステージに直接送信する場合、または、マッピングソフトウェアへの入力など、他のアプリケーションでルートを表示するために使用する場合のみ、このオプションを選択してください。</p>
CoordinateSystem	<p>入力される座標の座標系。</p> <p>EPSG コードの詳細については、<a href="http://www.spatialreference.org">www.spatialreference.org</a> を参照してください。EPSG に対してサポートされるコードスペースのリストを取得するには、SOAP リクエスト List Supported CoordSys by Code Space を Geometry サービス デモ ページ <a href="http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html">http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html</a> から送信します。</p>

Parameter	説明										
CoordinateFormat	<p>入力座標の緯度/経度のフォーマットを指定します。</p> <p>注: このオプションは、緯度/経度座標系を指定した場合のみに使用します。座標系が緯度/経度座標系ではない場合は、座標フォーマットを [小数] に設定します。</p> <p>次のいずれかです。</p> <table border="0"> <tr> <td><b>Decimal</b></td> <td>(90.000000, 180.000000)。デフォルト</td> </tr> <tr> <td><b>DecimalAssumed</b></td> <td>(90000000, 180000000)。</td> </tr> <tr> <td><b>DegMinSec</b></td> <td>(90 00 00N, 180 00 00W)</td> </tr> <tr> <td><b>PreZero</b></td> <td>(090000000N, 180000000W)</td> </tr> <tr> <td><b>PreZeroDecimal</b></td> <td>(090.000000N, 180.000000W)</td> </tr> </table>	<b>Decimal</b>	(90.000000, 180.000000)。デフォルト	<b>DecimalAssumed</b>	(90000000, 180000000)。	<b>DegMinSec</b>	(90 00 00N, 180 00 00W)	<b>PreZero</b>	(090000000N, 180000000W)	<b>PreZeroDecimal</b>	(090.000000N, 180.000000W)
<b>Decimal</b>	(90.000000, 180.000000)。デフォルト										
<b>DecimalAssumed</b>	(90000000, 180000000)。										
<b>DegMinSec</b>	(90 00 00N, 180 00 00W)										
<b>PreZero</b>	(090000000N, 180000000W)										
<b>PreZeroDecimal</b>	(090.000000N, 180.000000W)										
DistanceUnits	<p>GetTravelCostMatrix が返す距離の値の単位を指定します。次のいずれかです。</p> <table border="0"> <tr> <td><b>Feet</b></td> <td>距離をフィートで返します。</td> </tr> <tr> <td><b>Kilometers</b></td> <td>距離をキロメートルで返します。</td> </tr> <tr> <td><b>Meters</b></td> <td>距離をメートルで返します。</td> </tr> <tr> <td><b>Miles</b></td> <td>距離をマイルで返します。デフォルト</td> </tr> <tr> <td><b>Yards</b></td> <td>距離をヤードで返します。</td> </tr> </table>	<b>Feet</b>	距離をフィートで返します。	<b>Kilometers</b>	距離をキロメートルで返します。	<b>Meters</b>	距離をメートルで返します。	<b>Miles</b>	距離をマイルで返します。デフォルト	<b>Yards</b>	距離をヤードで返します。
<b>Feet</b>	距離をフィートで返します。										
<b>Kilometers</b>	距離をキロメートルで返します。										
<b>Meters</b>	距離をメートルで返します。										
<b>Miles</b>	距離をマイルで返します。デフォルト										
<b>Yards</b>	距離をヤードで返します。										
TimeUnits	<p>時間の返し方を指定します。次のいずれかです。</p> <table border="0"> <tr> <td><b>Hours</b></td> <td>時間を時間数で返します。</td> </tr> <tr> <td><b>Minutes</b></td> <td>時間を分数で返します。デフォルト</td> </tr> <tr> <td><b>Seconds</b></td> <td>時間を秒数で返します。</td> </tr> <tr> <td><b>Milliseconds</b></td> <td>時間をミリ秒単位で返します。</td> </tr> </table>	<b>Hours</b>	時間を時間数で返します。	<b>Minutes</b>	時間を分数で返します。デフォルト	<b>Seconds</b>	時間を秒数で返します。	<b>Milliseconds</b>	時間をミリ秒単位で返します。		
<b>Hours</b>	時間を時間数で返します。										
<b>Minutes</b>	時間を分数で返します。デフォルト										
<b>Seconds</b>	時間を秒数で返します。										
<b>Milliseconds</b>	時間をミリ秒単位で返します。										

## Parameter

## 説明

## HistoricTrafficTimeBucket

ルーティング計算において、旅行速度履歴を使用するかどうかを指定します。これらの速度は、1日のうちの異なる時間帯に基づきます。この機能を使用するには、旅行速度履歴がデータに含まれている必要があります。各国/地域のデータに対する時間帯定義は同じです。時間帯に対する速度は異なる場合があります。次のオプションがあります。

<b>None</b>	デフォルト値です。計算において、旅行速度履歴は使用しません。代わりに速度の平均値を使用します。
<b>AMPeak</b>	午前のピーク速度でルートを計算します。午前ピークの時間帯は、1日のうちの 07:00 ~ 10:00 です。
<b>PMPeak</b>	午後のピーク速度でルートを計算します。午後ピークの時間帯は、1日のうちの 16:00 ~ 19:00 です。
<b>OffPeak</b>	オフピーク (日中) 速度でルートを計算します。オフピークの時間帯は、1日のうちの 10:00 ~ 16:00 です。
<b>Night</b>	夜間速度でルートを計算します。夜間の時間帯は、1日のうちの 22:00 ~ 04:00 です。

## 一時オプション

この優先設定セットを使用して、要求ごとに一時更新を設定することができます。例えば、サーバーがすべての幹線道路タイプを回避するよう要求することができます。要求ごとに 1 つ以上の更新を含めることができます。

**注：**一時更新機能は SOAP API を介してのみ使用可能で、Management Console または Enterprise Designer を介しては使用できません。



表 85 : 一時更新オプション

パラメータ	説明
PointUpdate	<p>ポイントの更新は、対応するポイント (緯度、経度) に適用される変更です。特定のポイントに対して、ポイントの除外、ポイントの速度の設定、値または割合によるポイントの速度の変更 (増加または減少) を行うことができます。Point を Latitude と Longitude で指定してから、次のいずれかを指定する必要があります。</p>
<b>Velocity</b>	<p>速度単位と新しい速度を指定してポイントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p>
<b>SpeedIncrease</b>	<p>速度 (単位と値) または速度を増加させる割合を指定してポイントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p>
<b>SpeedDecrease</b>	<p>速度 (単位と値) または速度を減少させる割合を指定してポイントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p>
<b>Exclude</b>	<p>指定されたポイントをルート計算から除外する文字列値です。ポイントを除外するには、ポイントを指定して、Y と定義した Exclude パラメータを含める必要があります。有効な値は Y (はい) と N (いいえ) です。</p>

パラメータ

説明

---

SegmentUpdate

## パラメータ

### 説明

セグメントの更新は、対応するセグメント ID (緯度、経度) に適用される変更です。特定のセグメントに対して、セグメントの除外、セグメントの速度の設定、値または割合によるセグメントの速度の変更 (増加または減少)、セグメントの道路タイプの変更を行うことができます。有効な RoutingSegmentID を指定してから、次のいずれかを指定する必要があります。

**Velocity** 速度単位と新しい速度を指定してセグメントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には kph (キロメートル/時)、mph (マイル/時)、mps (メートル/秒)、mtpm (メートル/分) のいずれかの値を指定できます。

**SpeedIncrease** 速度 (単位と値) または速度を増加させる割合を指定してセグメントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には kph (キロメートル/時)、mph (マイル/時)、mps (メートル/秒)、mtpm (メートル/分) のいずれかの値を指定できます。

**SpeedDecrease** 速度 (単位と値) または速度を減少させる割合を指定してセグメントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には kph (キロメートル/時)、mph (マイル/時)、mps (メートル/秒)、mtpm (メートル/分) のいずれかの値を指定できます。

**RoadType** ルート計算に使用するセグメントの道路タイプの値を変更する文字列値です。

RoadType には、次の種類があります。

- access way
- back road
- connector
- ferry
- footpath
- limited access dense urban
- limited access rural
- limited access suburban
- limited access urban
- local road dense urban
- local road rural
- local road suburban
- local road urban
- major local road dense urban
- major local road rural
- major local road suburban
- major local road urban
- major road dense urban
- major road rural

## パラメータ

## 説明

- 
- major road suburban
  - major road urban
  - minor local road dense Urban
  - minor local road rural
  - minor local road suburban
  - minor local road urban
  - normal road dense urban
  - normal road rural
  - normal road rural
  - normal road urban
  - primary highway dense urban
  - primary highway rural
  - primary highway suburban
  - primary highway urban
  - ramp dense urban
  - ramp limited access
  - ramp major road
  - ramp primary highway
  - ramp rural
  - ramp secondary highway
  - ramp urban
  - ramp suburban
  - secondary highway dense urban
  - secondary highway rural
  - secondary highway suburban
  - secondary highway urban

**Exclude**

指定されたセグメントをルート計算から除外する文字列値です。セグメントを除外するには、セグメント ID を指定して、Y と定義した Exclude パラメータを含める必要があります。有効な値は Y (はい) と N (いいえ) です。

パラメータ	説明
-------	----

<b>RoadTypeUpdate</b>	<p>道路タイプの更新は、対応する道路タイプに適用される変更です。特定の道路タイプに対して、道路タイプの速度の設定、値または割合による道路タイプの速度の変更 (増加または減少) を行うことができます。更新する RoadType (セグメントの更新で前述した道路タイプを参照) を指定してから、次のいずれかを指定する必要があります。</p> <p><b>Velocity</b> 速度単位と新しい速度を指定してセグメントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p> <p><b>SpeedIncrease</b> 速度 (単位と値) または速度を増加させる割合を指定してセグメントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p> <p><b>SpeedDecrease</b> 速度 (単位と値) または速度を減少させる割合を指定してセグメントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p>
-----------------------	---

### GetTravelCostMatrix の一時更新の SOAP の例

使用可能な一時更新オプションをすべて使った標準的な **GetTravelCostMatrix** の SOAP 要求を以下に示します (すべての構文を示すためのもので、実例ではありません)。要求ごとに、各ルートマトリクスの計算に使用する **MatrixTransientUpdate** を 1 つ含めることができます。複数の **Update** 定義を 1 つの **MatrixTransientUpdate** に含めることができます。1 つの更新タイプ (**PointUpdate**、**SegmentUpdate**、または **RoadTypeUpdate**) だけを 1 つの **Update** に含めることができます。いずれかの更新タイプ (**PointUpdate**、**SegmentUpdate**、または **RoadTypeUpdate**) 内に含めることができる更新も 1 つだけです。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:get="http://www.pb.com/spectrum/services/GetTravelCostMatrix"

  xmlns:spec="http://spectrum.pb.com/"
  xmlns:get1="http://www.g1.com/services/GetTravelCostMatrix"
  xmlns:typ="http://www.g1.com/services/erm/types">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetTravelCostMatrixRequest>
      <get:input_port>
```

```

    <get:RouteMatrixRequest>
      <get:StartPoints>
        <get:StartPoint>
          <get:Latitude>33.751748</get:Latitude>

          <get:Longitude>-84.364014</get:Longitude>
        </get:StartPoint>
        <get:StartPoint>
          <get:Latitude>33.870416</get:Latitude>
          <get:Longitude>-78.62915</get:Longitude>
        </get:StartPoint>
        <get:StartPoint>
          <get:Latitude>35.025498</get:Latitude>
          <get:Longitude>-80.864868</get:Longitude>
        </get:StartPoint>
      </get:StartPoints>
      <get:EndPoints>
        <get:EndPoint>
          <get:Latitude>33.664925</get:Latitude>
          <get:Longitude>-80.90332</get:Longitude>
        </get:EndPoint>
        <get:EndPoint>
          <get:Latitude>34.40691</get:Latitude>
          <get:Longitude>-80.062866</get:Longitude>
        </get:EndPoint>
        <get:EndPoint>
          <get:Latitude>34.921971</get:Latitude>
          <get:Longitude>-81.013184</get:Longitude>
        </get:EndPoint>
      </get:EndPoints>
    </get:RouteMatrixRequest>
  </get:MatrixTransientUpdate>
  <typ:Update>
    <typ:PointUpdate>
      <typ:Point>
        <typ:Latitude>?</typ:Latitude>
        <typ:Longitude>?</typ:Longitude>
      </typ:Point>
      <typ:SpeedUpdate>
        <typ:Velocity VelocityUnit=""/>
        <typ:SpeedIncrease>
          <typ:Velocity VelocityUnit=""/>
          <typ:Percentage>?</typ:Percentage>
        </typ:SpeedIncrease>
        <typ:SpeedDecrease>
          <typ:Velocity VelocityUnit="">?</typ:Velocity>
          <typ:Percentage>?</typ:Percentage>
        </typ:SpeedDecrease>
      </typ:SpeedUpdate>
      <typ:Exclude>?</typ:Exclude>
    </typ:PointUpdate>
    <typ:SegmentUpdate>
      <typ:RoutingSegmentID>?</typ:RoutingSegmentID>
      <typ:SpeedUpdate>

```

```

    <typ:Velocity VelocityUnit="">?</typ:Velocity>
    <typ:SpeedIncrease>
    <typ:Velocity VelocityUnit="">?</typ:Velocity>
    <typ:Percentage>?</typ:Percentage>
  </typ:SpeedIncrease>
  <typ:SpeedDecrease>
  <typ:Velocity VelocityUnit="">?</typ:Velocity>
  <typ:Percentage>?</typ:Percentage>
</typ:SpeedDecrease>
</typ:SpeedUpdate>
<typ:RoadType>?</typ:RoadType>
<typ:Exclude>?</typ:Exclude>
</typ:SegmentUpdate>
<typ:RoadTypeUpdate>
<typ:RoadType>?</typ:RoadType>
<typ:SpeedUpdate>
<typ:Velocity VelocityUnit="">?</typ:Velocity>
<typ:SpeedIncrease>
<typ:Velocity VelocityUnit="">?</typ:Velocity>
<typ:Percentage>?</typ:Percentage>
  </typ:SpeedIncrease>
  <typ:SpeedDecrease>
  <typ:Velocity VelocityUnit="">?</typ:Velocity>
  <typ:Percentage>?</typ:Percentage>
</typ:SpeedDecrease>
</typ:SpeedUpdate>
</typ:RoadTypeUpdate>
</typ:Update>
</get:MatrixTransientUpdate>
  </get:RouteMatrixRequest>
  </get:input_port>
  </get:GetTravelCostMatrixRequest>
</soapenv:Body>
</soapenv:Envelope>

```

## 応答

**GetTravelCostMatrix** からの出力には多数のルート シーケンスが含まれます。各ルート シーケンスには、マトリクスルートで使用される候補の始点と終点、および計算された各ルートの時間と距離が含まれます。**OptimizeBy** オプションの設定によって、出力にマトリクスのすべてのルートを含めることも、各始点/終点間の最適ルートのみを含めることもできます。

**GetTravelCostMatrix** でマトリクスの始点と終点の間にルートが見つからないと、サーバー ログにエラーが記録されますが、確認できたルートはすべて返されます。例えば、始点 A と終点 1 および 2 があって、**GetTravelCostMatrix** が A から 1 へのルートは見つけれられても、A から 2 へのルートは見つけれられない場合、**GetTravelCostMatrix** は A から 1 へのルートを返し、A から 2 へのルートは見つけれなかったことをログに記録します。

**GetTravelCostMatrix** の出力は、**ReturnRouteCostMatrix** オプションの設定に応じて、リスト形式かオブジェクト形式になります。オブジェクト出力を直接操作することはできません。

表 86 : Get Travel Cost Matrix の出力

応答要素	書式	説明
Distance	文字列	始点から終点までの距離。値の単位は、DistanceUnits 要素に設定された単位になります。
DistanceUnits	文字列	距離に使用される単位です。次のいずれかです。 <b>Feet</b> 距離はフィート単位です。 <b>Kilometers</b> 距離はキロメートル単位です。 <b>Meters</b> 距離はメートル単位です。 <b>Miles</b> 距離はマイル単位です。デフォルト <b>Yards</b> 距離はヤード単位です。
EndPointRef	文字列	入力で終点が指定された順序に対応する参照 ID。最初に指定された終点の参照 ID は 1、2 番目に指定された終点の参照 ID は 2、以降も同様になります。入力の緯度/経度座標を Get Travel Cost Matrix が返す参照 ID と関連付ける独自プロセスを作成する必要があります。
EndPointID	文字列	Input ステージの ID フィールドで、対応する終점에割り当てた ID。例えば、最初の終点の EndPointID が N、2 つめの終点の EndPointID が O (以下同様) となっている場合があります。  注：このフィールドが有効になるのは、[ルート コスト マトリクス フォーマット] フィールドが [階層] に設定されている場合だけです。
StartPointRef	文字列	入力で始点が指定された順序に対応する参照 ID。最初に指定された始点の参照 ID は 1、2 番目に指定された始点の参照 ID は 2、以降も同様になります。入力の緯度/経度座標を Get Travel Cost Matrix が返す参照 ID と関連付ける独自プロセスを作成する必要があります。



応答要素	書式	説明
StartPointID	文字列	<p>Input ステージの ID フィールドにおいて対応する始点に割り当てた識別子。例えば、最初の始点の StartPointID が A、2 つめの始点の StartPointID が B (以下同様) となっている場合があります。</p> <p>注：このフィールドが有効になるのは、<b>[ルート コスト マトリクス フォーマット]</b> フィールドが <b>[階層]</b> に設定されている場合だけです。</p>
Time	文字列	始点から終点までの合計時間。値の単位は、TimeUnits 要素に設定された単位になります。
TimeUnits	文字列	<p>時間に使用される単位です。次のいずれかです。</p> <p><b>Hours</b>                      単位は時間になります。</p> <p><b>Minutes</b>                    単位は分になります。デフォルト</p> <p><b>Seconds</b>                    単位は秒になります。</p> <p><b>Milliseconds</b>              単位はミリ秒になります。</p>
Status	文字列 [1]	<p>マッチが成功したか失敗したかを表します。</p> <p><b>NULL</b>                      成功</p> <p><b>F</b>                              失敗</p>
Status.Code	文字列 [100]	失敗の原因 (ある場合)。
Status.Description	文字列	Status.Code に示された失敗の説明です。

### GetTravelDirections (Legacy)

**重要：**このステージは 12.1 リリースで非推奨になっています。新しいデータフローを作成するときは、代わりに **ルート** ステージを使用してください。

`GetTravelDirections` は、2つのポイントまたは複数のポイントに対するルーティング情報を返します。始点の緯度/経度と終点の緯度/経度を入力として受け取り、ステージの設定方法によって、最速または最短ルートを返します。

国ごとに、`Enterprise Routing` モジュール-<国> という形式の名前が付けられた固有のデータベースがあります。各データベースには、固有の国コードも付与されています。例えば、オーストリアのデータベースの名前は、"`Enterprise Routing` モジュール-オーストリア"で、オーストリアのバッチ国コードは "`A1T`" です。データベースごとに個別のライセンスが必要です。

**注：** `Get Travel Directions` は、SOAP Web サービスとしてのみ利用可能です。`Get Travel Directions` を、REST を介して利用することはできません。Java、C++、C、.NET、または COM API を介して利用することもできません。

`GetTravelDirections` は、`Enterprise Routing` モジュールに含まれています。

### リソース URL

```
http://server:port/soap/GetTravelDirections
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetTravelDirections"
xmlns:typ="http://www.g1.com/services/erm/types">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetTravelDirectionsRequest>
      <get:input_port>
        <get:PointToPointRequest>
          <get:RoutePoints>
            <get:RoutePoint>
              <get:Latitude>33.751748</get:Latitude>

              <get:Longitude>-84.364014</get:Longitude>
            </get:RoutePoint>
            <get:RoutePoint>
              <get:Latitude>33.664925</get:Latitude>
              <get:Longitude>-80.90332</get:Longitude>
            </get:RoutePoint>
          </get:RoutePoints>
        </get:PointToPointRequest>
      </get:input_port>
    </get:GetTravelDirectionsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

注：この例では、短縮のため一部の道順が削除されています。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns6:GetTravelDirectionsResponse
xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.g1.com/services/erm/types"
xmlns:ns4="http://www.g1.com/services/GetTravelDirections"
xmlns:ns5="http://www.mapinfo.com/midev/service/geometries/v1"
xmlns:ns6="http://www.pb.com/spectrum/services/GetTravelDirections">
      <ns6:output_port>
        <ns6:PointToPointResponse>
          <ns6:Time>215.82</ns6:Time>
          <ns6:TimeUnits>Minutes</ns6:TimeUnits>
          <ns6:Distance>218.441</ns6:Distance>
          <ns6:DistanceUnits>Miles</ns6:DistanceUnits>
          <ns6:Format>Normal</ns6:Format>
          <ns6:Language>en</ns6:Language>
          <ns6:RouteDirections>
            <ns6:RouteDirection>
              <ns6:Instruction/>
              <ns6:Time>0.03</ns6:Time>
              <ns6:TimeUnits>Minutes</ns6:TimeUnits>
              <ns6:Distance>0.013</ns6:Distance>
              <ns6:DistanceUnits>Miles</ns6:DistanceUnits>
            </ns6:RouteDirection>
            <ns6:RouteDirection>
              <ns6:Instruction>Turn left on Short St SE and travel
South 0.10 mi (0.3 min).</ns6:Instruction>
              <ns6:Time>0.28</ns6:Time>
              <ns6:TimeUnits>Minutes</ns6:TimeUnits>
              <ns6:Distance>0.099</ns6:Distance>
              <ns6:DistanceUnits>Miles</ns6:DistanceUnits>
            </ns6:RouteDirection>
            ...
            <ns6:RouteDirection>
              <ns6:Instruction>Turn left on Un-named street and
travel East 0.11 mi (0.2 min).</ns6:Instruction>
              <ns6:Time>0.2</ns6:Time>
              <ns6:TimeUnits>Minutes</ns6:TimeUnits>
              <ns6:Distance>0.105</ns6:Distance>
              <ns6:DistanceUnits>Miles</ns6:DistanceUnits>
            </ns6:RouteDirection>
            <ns6:RouteDirection>
              <ns6:Instruction>Turn right to reach your
destination to the East.</ns6:Instruction>
              <ns6:Time>0.33</ns6:Time>
              <ns6:TimeUnits>Minutes</ns6:TimeUnits>
              <ns6:Distance>0.167</ns6:Distance>
              <ns6:DistanceUnits>Miles</ns6:DistanceUnits>
            </ns6:RouteDirection>
          </ns6:RouteDirections>
        </ns6:PointToPointResponse>
      </ns6:output_port>
    </ns6:GetTravelDirectionsResponse>
  </soap:Body>
</soap:Envelope>
```

```

        </ns6:RouteDirection>
      </ns6:RouteDirections>
    </ns6:user_fields/>
  </ns6:PointToPointResponse>
</ns6:output_port>
</ns6:GetTravelDirectionsResponse>
</soap:Body>
</soap:Envelope>

```

### 要求

#### 入力データのパラメータ

**GetTravelDirections** は、始点と終点の緯度/経度座標を含むルート ポイントを入力として受け取ります。以下の表に、入力のフォーマットとレイアウトに関する情報を示します。

**注：** **GetTravelDirections** サービスを操作する場合は、これが Web サービスとしてのみ利用可能であることに注意してください。 **Get Travel Directions** サービスを、Java、C++、C、.NET、または COM API を介して利用することはできません。

**表 87 : GetTravel Directions の入力データ**

パラメータ	書式	説明
RoutePoints	リスト	緯度と経度の文字列が含まれるスキーマです。 プロセスリストの作成については、『 <i>Spectrum™ Technology Platform データフロー デザイナー ガイド</i> 』を参照してください。

パラメータ	書式	説明
-------	----	----

---

Language	文字列	
----------	-----	--

パラメータ	書式	説明
-------	----	----

`GetTravelDirections` が道順を返す言語を指定します。

次のいずれかです。

<b>sq</b>	アルバニア語で道順を返します。
<b>zh_CN</b>	中国語で道順を返します。
<b>zh_TW</b>	中国語 (台湾) で道順を返します。
<b>hr</b>	クロアチア語で道順を返します。
<b>cs</b>	チェコ語で道順を返します。
<b>da</b>	デンマーク語で道順を返します。
<b>nl</b>	オランダ語で道順を返します。
<b>en</b>	英語で道順を返します。デフォルト
<b>en-US</b>	アメリカ英語で道順を返します。
<b>et</b>	エストニア語で道順を返します。
<b>fi</b>	フィンランド語で道順を返します。
<b>fr</b>	フランス語で道順を返します。
<b>de</b>	ドイツ語で道順を返します。
<b>hu</b>	ハンガリー語で道順を返します。
<b>it</b>	イタリア語で道順を返します。
<b>ja</b>	日本語で道順を返します。
<b>lv</b>	ラトビア語で道順を返します。
<b>lt</b>	リトアニア語で道順を返します。
<b>no</b>	ノルウェー語で道順を返します。
<b>pt</b>	ポルトガル語で道順を返します。
<b>ro</b>	ルーマニア語で道順を返します。
<b>sk</b>	スロバキア語で道順を返します。
<b>sl</b>	スロベニア語で道順を返します。
<b>es</b>	スペイン語で道順を返します。
<b>sv</b>	スウェーデン語で道順を返します。
<b>ru</b>	ロシア語で道順を返します。
<b>tr</b>	トルコ語で道順を返します。

パラメータ	書式	説明
-------	----	----

注：この入力フィールドに入力した値は、[デフォルト言語] オプションの値に優先して適用されます。

TravelDirectionTransientUpdate	リスト	<p>一時更新の更新タイプを含むスキーマです。一時更新は、特定の要求にのみ適用される更新です。一時更新は永続更新に似ていますが、一時更新は特定の要求のみに対応し、永続更新はすべての要求に対応しています。ポイントの速度、セグメント ID、または道路クラスを設定できるだけでなく、セグメント (セグメント ID で指定) の道路クラスを更新できます。</p> <p>一時更新のオプションと例は、<a href="#">GetTravelDirections_TransientOptions.dita</a>を参照してください。</p>
--------------------------------	-----	--

TollRoad	Boolean	<p>この機能では、ルートに有料道路を含めるかどうかを指定できます。これは Boolean 型のパラメータです。デフォルト値は False です。TollRoad の値を True に設定した場合、レスポンスには有料道路が一切含まれていないルートが格納されます。TollRoad の値が False に設定されている場合、ルートには有料道路が含まれます。</p>
----------	---------	---

### オプションのパラメータ ルーティング

以下の表に、GetTravelDirections の構成オプションを示します。

**表 88 : Get Travel Directions の構成オプション**

パラメータ	説明
DataSetResourceName	<p>検索プロセスで使用するデータが格納されたデータベースの名前。Management Console の [リソース] セクションで定義した有効なルーティング データベース リソース名を使用します。詳細については、『<i>Spectrum™ Technology Platform Spatial ガイド</i>』を参照してください。</p>

パラメータ	説明
OptimizeBy	<p>GetTravelDirections が最短距離または最短時間のどちらを検索するかを指定します。</p> <p>次のいずれかです。</p> <p><b>Time</b> 移動時間が最短となるように最適化します。デフォルト</p> <p><b>Distance</b> 移動距離が最短となるように最適化します。</p>
CoordinateSystem	<p>入力される座標の座標系。</p> <p>EPSG コードの詳細については、<a href="http://www.spatialreference.org">www.spatialreference.org</a> を参照してください。EPSG に対してサポートされるコードスペースのリストを取得するには、SOAP リクエスト List Supported CoordSys by Code Space を Geometry サービス デモ ページ <a href="http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html">http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html</a> から送信します。</p>
CoordinateFormat	<p>入力座標の緯度/経度のフォーマットを指定します。</p> <p>注：このオプションは、緯度/経度座標系を指定した場合のみに使用します。座標系が緯度/経度座標系ではない場合は、座標フォーマットを [小数] に設定します。</p> <p>次のいずれかです。</p> <p><b>Decimal</b> (90.000000, 180.000000)。デフォルト</p> <p><b>DecimalAssumed</b> (90000000, 180000000)。</p> <p><b>DegreesMinutesSeconds</b> (90 00 00N, 180 00 00W)</p> <p><b>PreZero</b> (090000000N, 180000000W)</p> <p><b>PreZeroDecimal</b> (090.000000N, 180.000000W)</p>
DirectionsStyle	<p>道順を返すフォーマットを指定します。</p> <p>次のいずれかです。</p> <p><b>Normal</b> 標準フォーマットで道順を返します。デフォルト</p> <p><b>Terse</b> 簡易フォーマットで道順を返します。簡易フォーマットの道順は、ワイヤレス機器に適しています。</p>



## パラメータ

## 説明

---

**DistanceUnits**

`GetTravelDirections` が返す距離の値の単位を指定します。

次のいずれかです。

<b>Feet</b>	距離をフィートで返します。
<b>Kilometers</b>	距離をキロメートルで返します。
<b>Meters</b>	距離をメートルで返します。
<b>Miles</b>	距離をマイルで返します。デフォルト
<b>Yards</b>	距離をヤードで返します。

---

**TimeUnits**

`GetTravelDirections` が返す時間の値の単位を指定します。

次のいずれかです。

<b>Hours</b>	時間を時間数で返します。
<b>Minutes</b>	時間を分数で返します。デフォルト
<b>Seconds</b>	時間を秒数で返します。
<b>Milliseconds</b>	時間をミリ秒単位で返します。

---

**FocusOfRoute**

ルートの優先項目を指定します。優先項目が指定されたルートは、ルートの始点または終点のいずれかに着目した、ルート全体のサブセットになります。始点を優先項目とするルートは、始点から最初の主要な高速道路まで(その高速道路を含む)のルートを示します。終点を優先項目とするルートは、ルートの中の最後の主要な高速道路(その高速道路を含む)から目的地までのルートを示します。ルートに主要な高速道路がない場合、優先項目が指定されたルートは優先項目のないルートと同じになります。

次のいずれかです。

<b>Start</b>	ルートの最初のみを返します。
<b>End</b>	ルートの最後のみを返します。
<b>None</b>	ルート全体を返します。デフォルト

---

## パラメータ

## 説明

## HistoricTrafficTimeBucket

ルーティング計算において、旅行速度履歴を使用するかどうかを指定します。これらの速度は、1日のうちの異なる時間帯に基づきます。この機能を使用するには、旅行速度履歴がデータに含まれている必要があります。各国/地域のデータに対する時間帯定義は同じです。時間帯に対する速度は異なる場合があります。次のオプションがあります。

- None**      デフォルト値です。計算において、旅行速度履歴は使用しません。代わりに速度の平均値を使用します。
- AMPeak**    午前のピーク速度でルートを計算します。午前ピークの時間帯は、1日のうちの 07:00 ~ 10:00 です。
- PMPeak**    午後のピーク速度でルートを計算します。午後ピークの時間帯は、1日のうちの 16:00 ~ 19:00 です。
- OffPeak**    オフピーク (日中) 速度でルートを計算します。オフピークの時間帯は、1日のうちの 10:00 ~ 16:00 です。
- Night**      夜間速度でルートを計算します。夜間の時間帯は、1日のうちの 22:00 ~ 04:00 です。

## OptimizeIntermediatePoints

GetTravelDirections で、ルート計算時に通過点を最適な順序に並べ替えるかどうかを指定します。

- Y**      通過点を最適化します。道順において、通過点を最適な順序に並べ替えます。デフォルト
- N**      通過点を最適化しません。道順の通過点を指定された順序のままにします。

## localRoadsLoadFactor

ルート計算またはマトリクス計算中にメモリにロードできる地方道路の数を指定します。ロードできる道路の数は、このパラメータで選択した値に正比例します。パラメータの最小値は 1 で、最大値は 3 です。有効な値は 1、2、3 です。デフォルトは 1 です。パラメータがルート計算またはマトリクス計算に与える影響の詳細については「[地方道路の負荷係数](#)」を参照してください。

注: パラメータに小数の値を指定することはできません。

## 道順

表 89 : Get Travel Directions 道順オプション

パラメータ	説明
ShowDistance	ルートの距離を返すかどうかを指定します。
	<b>Y</b> ルートの距離を返します。デフォルト
	<b>N</b> ルートの距離を返しません。
ReturnRouteDirections	ルートの道順を進路変更ごとに示すかどうかを指定します。このオプションは、デフォルトで有効になっています。
	<b>Y</b> 進路変更ごとのテキスト表示の道順を返します。デフォルト
	<b>N</b> 進路変更ごとのテキスト表示の道順を返しません。
ReturnSegmentGeometry	ルート内のセグメントのジオメトリを表す緯度/経度ポイントの集合を返すかどうかを指定します。セグメント ジオメトリは、ルート ジオメトリを作成するために使用します。このフィールドの出力は、RouteDirections 出力フィールド内に表示されます。ルートジオメトリの詳細については、 <a href="#">ルートジオメトリとは</a> (493ページ) を参照してください。
ReturnRouteGeometry	ルート ジオメトリを表す緯度/経度ポイントの集合を返すかどうかを指定します。ルート ジオメトリは、ルート マップの作成およびルートに関する分析の実行に使用できます。
	次のいずれかです。
	<b>All</b> ルート ジオメトリのすべてのポイントを返します。
	<b>End</b> 各ルート セグメント ジオメトリの終点のみを返します。
<b>None</b> ルート ジオメトリを返しません。デフォルト	

## パラメータ

## 説明

## DefaultLanguage

GetTravelDirections が道順を返す言語を指定します。

次のいずれかです。

<b>sq</b>	アルバニア語で道順を返します。
<b>zh_CN</b>	中国語で道順を返します。
<b>zh_TW</b>	中国語 (台湾) で道順を返します。
<b>hr</b>	クロアチア語で道順を返します。
<b>cs</b>	チェコ語で道順を返します。
<b>da</b>	デンマーク語で道順を返します。
<b>nl</b>	オランダ語で道順を返します。
<b>en</b>	英語で道順を返します。デフォルト
<b>en-US</b>	アメリカ英語で道順を返します。
<b>et</b>	エストニア語で道順を返します。
<b>fi</b>	フィンランド語で道順を返します。
<b>fr</b>	フランス語で道順を返します。
<b>de</b>	ドイツ語で道順を返します。
<b>hu</b>	ハンガリー語で道順を返します。
<b>it</b>	イタリア語で道順を返します。
<b>ja</b>	日本語で道順を返します。
<b>lv</b>	ラトビア語で道順を返します。
<b>lt</b>	リトアニア語で道順を返します。
<b>no</b>	ノルウェー語で道順を返します。
<b>pt</b>	ポルトガル語で道順を返します。
<b>ro</b>	ルーマニア語で道順を返します。
<b>sk</b>	スロバキア語で道順を返します。
<b>sl</b>	スロベニア語で道順を返します。
<b>es</b>	スペイン語で道順を返します。
<b>sv</b>	スウェーデン語で道順を返します。
<b>ru</b>	ロシア語で道順を返します。
<b>tr</b>	トルコ語で道順を返します。

パラメータ	説明
ShowTime	<p>GetTravelDirections がルート内で道順をたどるのに要する時間を返すかどうかを指定します。</p> <p><b>Y</b>      ルートの時間を返します。デフォルト</p> <p><b>N</b>      ルートの時間を返しません。</p>
ShowPrimaryNameOnly	<p>GetTravelDirections が道順の中の任意のストリートのすべての名前を返すか、ストリートの主要名のみを返すかを指定します。</p> <p><b>Y</b>      主要名のみを返します。</p> <p><b>N</b>      通りのすべての名前を返します。デフォルト</p>

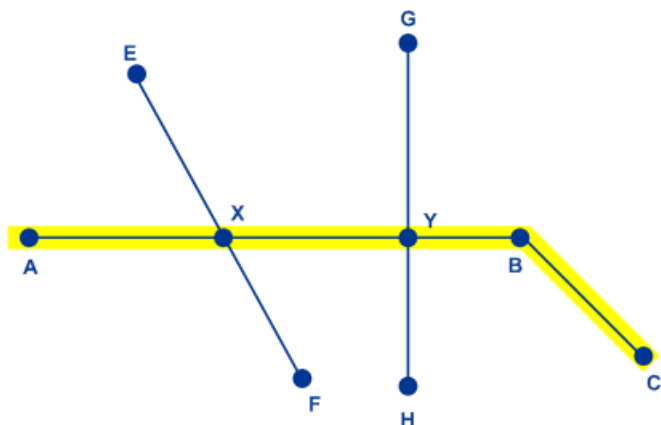
### ルート ジオメトリとは

ルート ジオメトリは、ルートを形成する一連の緯度/経度ポイントです。最も単純なルート ジオメトリは、単一のポイントです。同じ通りセグメント上で始まって終わるルートなどがこれに該当します。



始点は常に既知の場所なので、このように非常に単純なルート ジオメトリはすなわち終点に等しくなります。したがって、A が始点だとすると、ルート ジオメトリはポイント B の緯度/経度です。

複数のルート セグメントを含む、もっと複雑なルート ジオメトリの場合は、複数のポイントがルート ジオメトリ内に存在することがあります。以下に強調して示したようなルートがあると仮定します。ポイント A に始まり、交差点 X、Y、および B を経由してポイント C が終点となります。



このルートでは、完全なルート ジオメトリはポイント A、X、Y、B、および C の緯度/経度から構成されます。ただし、どのポイントを実際に返すかは、任意に制御できます。ルート ジオメトリのすべてのポイントを含めるのか、各ルート セグメントの終点のみを含めるのか、任意に選択できます。上記の例では、終点は B と C です。A から B は 1 つのルート セグメントであり、B から C は別のルート セグメントだからです。

#### 経路

この優先設定セットを使用して、道路タイプごとに望ましさを設定することができます。例えば、サーバーがすべての幹線道路タイプを回避するよう要求することができます。

表 90 : 経路の優先設定オプション

パラメータ	説明
-------	----

---

RoadType_<type>	
-----------------	--

## パラメータ

## 説明

ルート決定時にさまざまな道路に与える優先順位を指定します。

- access way
- back road
- connector
- ferry
- footpath
- limited access dense urban
- limited access rural
- limited access suburban
- limited access urban
- local road dense urban
- local road rural
- local road suburban
- local road urban
- major local road dense urban
- major local road rural
- major local road suburban
- major local road urban
- major road dense urban
- major road rural
- major road suburban
- major road urban
- minor local road dense Urban
- minor local road rural
- minor local road suburban
- minor local road urban
- normal road dense urban
- normal road rural
- normal road rural
- normal road urban
- primary highway dense urban
- primary highway rural
- primary highway suburban
- primary highway urban
- ramp dense urban
- ramp limited access
- ramp major road
- ramp primary highway
- ramp rural
- ramp secondary highway
- ramp urban
- ramp suburban
- secondary highway dense urban
- secondary highway rural



## パラメータ

## 説明

- secondary highway suburban
- secondary highway urban

道路タイプごとに、次のいずれかを指定できます。

**Avoid** この道路タイプを可能な限りルートで使用しません。

注：特定の道路タイプを道順から常に除外できるとは限りません。状況によっては、回避対象の道路タイプの代わりになるものが非常に貧弱なために回避対象の道路タイプを使うルートが選ばれることがあります。また、始点または終点がセグメントの横にあって、そのセグメントの道路タイプが回避されている場合も、そのセグメントが使われます。

**High** この道路タイプを他の道路タイプよりも優先します。

**Low** 他の道路タイプをこの道路タイプよりも優先します。

**Medium** この道路タイプに他の道路タイプと同じ優先設定を与えます。道路タイプに優先設定を指定しないと、デフォルトとして "中" が使われます。

パラメータ

説明

---

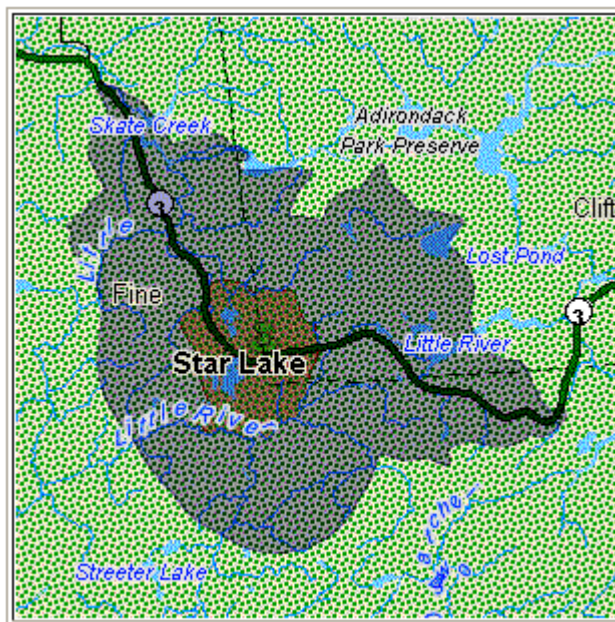
MajorRoads

## パラメータ

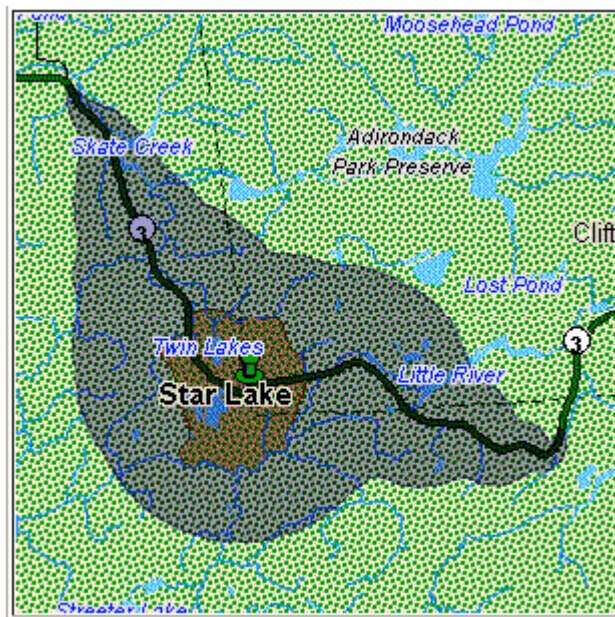
## 説明

計算時に、すべての道路を含めるか、幹線道路のみを含めるかを指定します。幹線道路のみを含めるように指定した場合は、パフォーマンスは向上しますが精度は低くなる場合があります。

次のマップは、すべての道路を走行する場合の経路境界を表します。



次のマップは、幹線道路のみを走行する場合の経路境界を表します。



## パラメータ

## 説明

---

次のいずれかです。

- Y** 計算時に、幹線道路のみを含めます。デフォルト
  - N** 計算時に、すべての道路を含めます。
-

パラメータ

説明

---

除外

有料道路

## パラメータ

## 説明

ルートに有料道路を含めるかどうかを指定できます。GetTravelCostMatrix GetTravelDirections ステージには、「有料道路の回避」機能が含まれています。UIに [有料道路] というラベルの付いたチェックボックスがあります。このチェックボックスをオンにすると、有料道路を回避できます。入力値で "TollRoad" としてこのパラメータを追加またはエクスポートすることもできます。入力値には Boolean 値を含むことができます。デフォルト値は False です。

**例:**

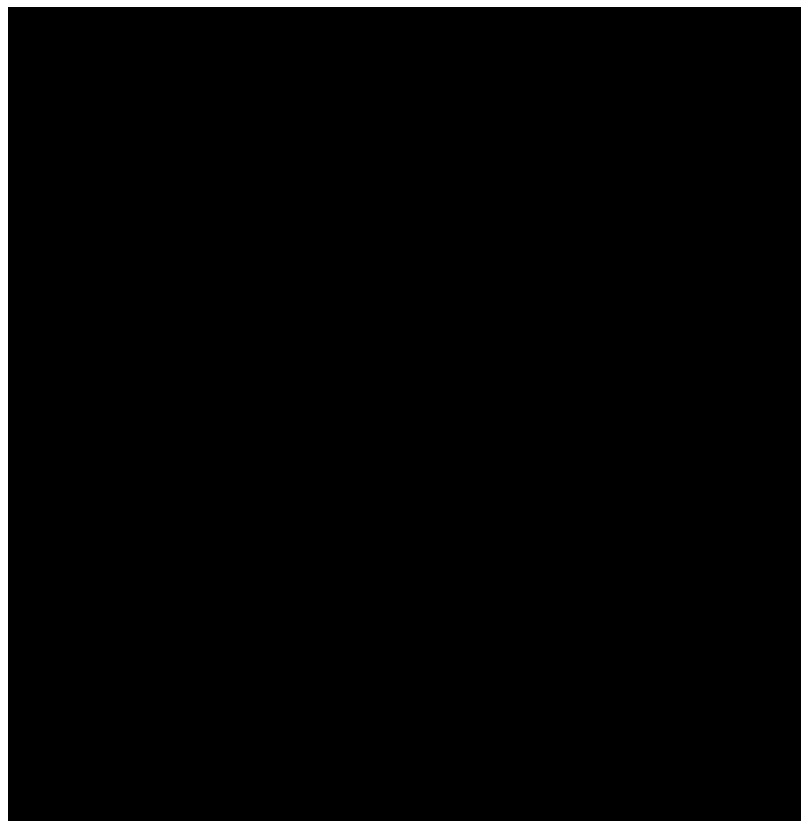
以下のルートには、有料道路情報が含まれています。以下の図を参照してください。



同じポイントに対して、[有料道路] チェック ボックスをオンにするか、または "TollRoad" パラメータの値を True に設定した場合、レスポンスには有料道路が一切含まれていないルートが格納されます。次のレスポンスを参照してください。

パラメータ

説明



---

#### 一時オプション

この優先設定セットを使用して、要求ごとに一時更新を設定することができます。例えば、サーバーがすべての幹線道路タイプを回避するよう要求することができます。要求ごとに1つ以上の更新を含めることができます。

**注：**一時更新機能は SOAP API を介してのみ使用可能で、Management Console または Enterprise Designer を介しては使用できません。

表 91 : 一時更新オプション

パラメータ	説明
PointUpdate	<p>ポイントの更新は、対応するポイント (緯度、経度) に適用される変更です。特定のポイントに対して、ポイントの除外、ポイントの速度の設定、値または割合によるポイントの速度の変更 (増加または減少) を行うことができます。Point を Latitude と Longitude で指定してから、次のいずれかを指定する必要があります。</p>
<b>Velocity</b>	<p>速度単位と新しい速度を指定してポイントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p>
<b>SpeedIncrease</b>	<p>速度 (単位と値) または速度を増加させる割合を指定してポイントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p>
<b>SpeedDecrease</b>	<p>速度 (単位と値) または速度を減少させる割合を指定してポイントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、<b>mph</b> (マイル/時)、<b>mps</b> (メートル/秒)、<b>mtpm</b> (メートル/分) のいずれかの値を指定できます。</p>
<b>Exclude</b>	<p>指定されたポイントをルート計算から除外する文字列値です。ポイントを除外するには、ポイントを指定して、Y と定義した Exclude パラメータを含める必要があります。有効な値は Y (はい) と N (いいえ) です。</p>



パラメータ

説明

---

SegmentUpdate

## パラメータ

## 説明

セグメントの更新は、対応するセグメント ID (緯度、経度) に適用される変更です。特定のセグメントに対して、セグメントの除外、セグメントの速度の設定、値または割合によるセグメントの速度の変更 (増加または減少)、セグメントの道路タイプの変更を行うことができます。有効な RoutingSegmentID を指定してから、次のいずれかを指定する必要があります。

**Velocity** 速度単位と新しい速度を指定してセグメントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

**SpeedIncrease** 速度 (単位と値) または速度を増加させる割合を指定してセグメントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

**SpeedDecrease** 速度 (単位と値) または速度を減少させる割合を指定してセグメントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

**RoadType** ルート計算に使用するセグメントの道路タイプの値を変更する文字列値です。

RoadType には、次の種類があります。

- access way
- back road
- connector
- ferry
- footpath
- limited access dense urban
- limited access rural
- limited access suburban
- limited access urban
- local road dense urban
- local road rural
- local road suburban
- local road urban
- major local road dense urban
- major local road rural
- major local road suburban
- major local road urban
- major road dense urban
- major road rural

## パラメータ

## 説明

- 
- major road suburban
  - major road urban
  - minor local road dense Urban
  - minor local road rural
  - minor local road suburban
  - minor local road urban
  - normal road dense urban
  - normal road rural
  - normal road rural
  - normal road urban
  - primary highway dense urban
  - primary highway rural
  - primary highway suburban
  - primary highway urban
  - ramp dense urban
  - ramp limited access
  - ramp major road
  - ramp primary highway
  - ramp rural
  - ramp secondary highway
  - ramp urban
  - ramp suburban
  - secondary highway dense urban
  - secondary highway rural
  - secondary highway suburban
  - secondary highway urban

**Exclude**

指定されたセグメントをルート計算から除外する文字列値です。セグメントを除外するには、セグメント ID を指定して、Y と定義した Exclude パラメータを含める必要があります。有効な値は Y (はい) と N (いいえ) です。

## パラメータ

## 説明

### RoadTypeUpdate

道路タイプの更新は、対応する道路タイプに適用される変更です。特定の道路タイプに対して、道路タイプの速度の設定、値または割合による道路タイプの速度の変更 (増加または減少) を行うことができます。更新する RoadType (セグメントの更新で前述した道路タイプを参照) を指定してから、次のいずれかを指定する必要があります。

**Velocity** 速度単位と新しい速度を指定してセグメントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

**SpeedIncrease** 速度 (単位と値) または速度を増加させる割合を指定してセグメントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

**SpeedDecrease** 速度 (単位と値) または速度を減少させる割合を指定してセグメントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には **kph** (キロメートル/時)、**mph** (マイル/時)、**mps** (メートル/秒)、**mtpm** (メートル/分) のいずれかの値を指定できます。

### *GetTravelDirections* の一時更新の SOAP の例

使用可能な一時更新オプションをすべて使った標準的な *GetTravelDirections* の SOAP 要求を以下に示します (すべての構文を示すためのもので、実例ではありません)。複数の Update 定義を 1 つの *TravelDirectionTransientUpdate* に含めることができます。1 つの更新タイプ (*PointUpdate*、*SegmentUpdate*、または *RoadTypeUpdate*) だけを 1 つの Update に含めることができます。いずれかの更新タイプ (*PointUpdate*、*SegmentUpdate*、または *RoadTypeUpdate*) 内に含めることができる更新も 1 つだけです。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetTravelDirections"
xmlns:typ="http://www.g1.com/services/erm/types">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetTravelDirectionsRequest>
      <get:input_port>
        <get:PointToPointRequest>
          <get:RoutePoints>
            <get:RoutePoint>
              <get:Latitude>33.751748</get:Latitude>
            </get:RoutePoint>
          </get:RoutePoints>
        </get:PointToPointRequest>
      </get:input_port>
    </get:GetTravelDirectionsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <get:Longitude>-84.364014</get:Longitude>
    </get:RoutePoint>
    <get:RoutePoint>
        <get:Latitude>33.664925</get:Latitude>
        <get:Longitude>-80.90332</get:Longitude>
    </get:RoutePoint>
</get:RoutePoints>
<get:Language>en-US</get:Language>
<get:TravelDirectionTransientUpdate>
    <typ:Update>
        <typ:PointUpdate>
            <typ:Point>
                <typ:Latitude>?</typ:Latitude>
                <typ:Longitude>?</typ:Longitude>
            </typ:Point>
            <typ:SpeedUpdate>
                <typ:Velocity VelocityUnit=""/>
                <typ:SpeedIncrease>
                    <typ:Velocity VelocityUnit=""/>
                    <typ:Percentage>?</typ:Percentage>
                </typ:SpeedIncrease>
                <typ:SpeedDecrease>
                    <typ:Velocity
VelocityUnit="">?</typ:Velocity>
                    <typ:Percentage>?</typ:Percentage>
                </typ:SpeedDecrease>
            </typ:SpeedUpdate>
            <typ:Exclude>?</typ:Exclude>
        </typ:PointUpdate>
        <typ:SegmentUpdate>

<typ:RoutingSegmentID>?</typ:RoutingSegmentID>
        <typ:SpeedUpdate>
            <typ:Velocity
VelocityUnit="">?</typ:Velocity>
            <typ:SpeedIncrease>
                <typ:Velocity
VelocityUnit="">?</typ:Velocity>
                <typ:Percentage>?</typ:Percentage>
            </typ:SpeedIncrease>
            <typ:SpeedDecrease>
                <typ:Velocity
VelocityUnit="">?</typ:Velocity>
                <typ:Percentage>?</typ:Percentage>
            </typ:SpeedDecrease>
        </typ:SpeedUpdate>
        <typ:RoadType>?</typ:RoadType>
        <typ:Exclude>?</typ:Exclude>
    </typ:SegmentUpdate>
<typ:RoadTypeUpdate>
    <typ:RoadType>?</typ:RoadType>
    <typ:SpeedUpdate>

```

```

    <typ:Velocity
VelocityUnit="?">?</typ:Velocity>
    <typ:SpeedIncrease>
VelocityUnit="?">?</typ:Velocity>
    <typ:Velocity
    <typ:Percentage?</typ:Percentage>
    </typ:SpeedIncrease>
    <typ:SpeedDecrease>
    <typ:Velocity
VelocityUnit="?">?</typ:Velocity>
    <typ:Percentage?</typ:Percentage>
    </typ:SpeedDecrease>
    </typ:SpeedUpdate>
    </typ:RoadTypeUpdate>
    </typ:Update>
    </get:TravelDirectionTransientUpdate>
    </get:PointToPointRequest>
  </get:input_port>
</get:GetTravelDirectionsRequest>
</soapenv:Body>
</soapenv:Envelope>

```

## 応答

GetTravelDirections は、次のフィールドを返します。

**表 92 : GetTravelDirections の出力**

応答要素	書式	説明
Distance	文字列	ルートに含まれる各セグメントの距離。
DistanceUnits	文字列	距離に使用される単位です。
Format	文字列	道順の生成に使用されるフォーマットの値です。
Language	文字列	道順の言語です。
RouteDirections	リスト	ルートの進路変更ごとの道順です。

応答要素	書式	説明
RouteGeometry	Geometry オブジェクト	ルートの各ポイントの座標を含むジオメトリ オブジェクトです。詳細については、 <a href="#">ルート ジオメトリとは</a> (493ページ)
Status	文字列	マッチの成功または失敗。  <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	文字列	失敗の原因 (ある場合)。次のいずれかです。  <ul style="list-style-type: none"> <li>• <b>InsufficientInputData</b> (緯度/経度の欠落)</li> <li>• <b>MalformedInputData</b> (誤った入力フォーマット)</li> <li>• <b>InputOutOfRange</b> (範囲外の入力)</li> <li>• <b>EngineError</b> (エンジン由来のエラー)</li> </ul>
Status.Description	文字列	<b>Status.Code</b> で示される失敗に関する説明。
Time	文字列	ルートをたどるのに要する合計時間を分数で表します。
TimeUnits	文字列	時間に使用される単位です。

## GetRouteData

**GetRouteData** は、ポイントまたはセグメント ID のルート セグメント情報を返します。ポイントを指定すると、最も近いルート セグメントが返されます。セグメント ID を指定すると、そのセグメント ID に対応するルート セグメントが返されます。

**注：** **Get Route Data** は、サービス (Management Console および SOAP Web サービス) としてのみ利用可能です。**Get Route Data** は、ステージまたは REST API を介して利用することはできません。Java、C++、C、.NET、または COM API を介して利用することもできません。

**GetRouteData** は、Spatial モジュールに含まれています。

## リソース URL

```
http://server:port/soap/GetRouteData
```

## 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.g1.com/services/GetRouteData">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetRouteDataRequest>
      <get:options>
        <get:DataSetResourceName>US</get:DataSetResourceName>
        <get:CoordinateSystem>epsg:4326</get:CoordinateSystem>
      </get:options>
      <get:rows>
        <get:row>
          <get:RoutingData>
            <get:RouteDataPoint>
              <get:Longitude>-74.843</get:Longitude>
              <get:Lattitude>40.0077</get:Lattitude>
            </get:RouteDataPoint>
          </get:RoutingData>
        </get:row>
      </get:rows>
    </get:GetRouteDataRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

注：この例では、簡略化のため一部のセグメントが割愛されています。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetRouteDataResponse
xmlns="http://www.g1.com/services/GetRouteData">
      <rows>
        <row>
          <Segments>
            <SegmentDetails>
              <Segment>
                <RoutingSegmentID>b81740d3:4b3526</RoutingSegmentID>
                <SegmentData>
                  <PrimaryName>New Jersey Tpke S</PrimaryName>
                  <PrimaryNameLanguage>en</PrimaryNameLanguage>
```



```

        <AlternateNameList>
          <AlternateName>
            <Name>New Jersey Tpke S</Name>
            <Language>en</Language>
          </AlternateName>
        </AlternateNameList>
        <SegmentLength>8.397</SegmentLength>
        <SegmentLengthUnit>Miles</SegmentLengthUnit>

        <TimeTaken>7.866666666666666</TimeTaken>
        <TimeUnit>Minutes</TimeUnit>
        <TurnAngle>0.0</TurnAngle>
        <TurnAngleUnit>degree</TurnAngleUnit>
        <CompassDirection/>

    </speedOfTravel>64.01366022429013</speedOfTravel>
    <speedOfTravelUnit>Miles/hour</speedOfTravelUnit>
    <RoadType>primary highway rural</RoadType>
    <SegmentDirection>from_to</SegmentDirection>

    <StartJunctionType>Other</StartJunctionType>

    <EndJunctionType/>
    <IsRoundabout>false</IsRoundabout>
    <IsTollRoad>true</IsTollRoad>
    <PointsInSegment>
      <RouteDataPoint>
        <Longitude>-74.823861</Longitude>
        <Latitude>40.024421</Latitude>
      </RouteDataPoint>
      <RouteDataPoint>
        <Longitude>-74.824133</Longitude>
        <Latitude>40.024149</Latitude>
      </RouteDataPoint>
      ...
    </PointsInSegment>
  </SegmentData>
</Segment>
</SegmentDetails>
</Segments>
</row>
</rows>
</GetRouteDataResponse>
</soap:Body>
</soap:Envelope>

```

**要求**

## 入力用のパラメータ

**GetRouteData** は、ポイントの緯度/経度、またはルート セグメント ID を入力として受け取ります。また、ルートデータソースとルートデータ座標系を指定する必要があります。以下の表に、入力のフォーマットとレイアウトに関する情報を示します。

**表 93 : GetRouteData の入力データ**

パラメータ	説明
RouteDataPoint	セグメント情報を返すためにクエリするポイント。 <b>Latitude</b> と <b>Longitude</b> から構成されるポイントを指定する必要があります。
RoutingSegmentID	セグメント情報を返すルート セグメント ID。
DataSetResourceName	セグメント情報をクエリするためのデータを格納したデータベースの名前。 <b>Spatial</b> モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。詳細については、『 <i>Spectrum™ Technology Platform 管理ガイド</i> 』を参照してください。
CoordinateSystem	入力される座標の座標系。  EPSG コードの詳細については、 <a href="http://www.spatialreference.org">www.spatialreference.org</a> を参照してください。EPSG に対してサポートされるコードスペースのリストを取得するには、SOAP リクエスト <b>List Supported CoordSys by Code Space</b> を <b>Geometry サービス デモ ページ</b> <a href="http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html">http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html</a> から送信します。

## オプション

表 94 : Get Route Data オプション

パラメータ	説明
CoordinateFormat	応答で返されるポイント データの座標のフォーマットを指定する文字列値。使用できるオプションは、Decimal、DecimalAssumed、PreZero、PreZeroDecimal、および DegMinSec です。
DistanceUnits	応答で返されるセグメント距離値 (セグメントの長さ) のフォーマットを指定する文字列合値。使用できるオプションは、Feet、Yards、Miles、Meters、および Kilometers です。
TimeUnits	応答で返されるセグメント時間値 (セグメントを通過する時間) のフォーマットを指定する文字列合値。使用できるオプションは、Milliseconds、Seconds、Minutes、および Hours です。
VelocityUnits	応答で返されるセグメント速度値のフォーマットを指定する文字列合値。使用できるオプションは、KPH (キロメートル/時)、MPH (マイル/時)、MTPS (メートル/秒)、および MTPM (メートル/分) です。
AngularUnits	応答で返されるセグメント折返し値 (角度単位) のフォーマットを指定する文字列合値。使用できるオプションは、radian、degree、minute、second、および grad です。
ReturnActualPreferences	応答でルート優先設定のリストを返すかどうかを指定する文字列合値。使用できるオプションは、Y または N です。デフォルトでは、ルート優先設定は返しません。
ReturnSegmentGeometry	セグメント ジオメトリについて返す情報の詳細レベルを指定する文字列値。使用できるオプションは、NONE、ALL、または END です。ALL を指定すると、セグメントジオメトリ全体が返されます。END を指定すると、セグメントの終点のみが返されます。デフォルトは、セグメント ジオメトリ全体を返す ALL です。

## 応答

### 応答

GetRouteData は、次のフィールドを返します。

表 95 : GetRouteData の出力

応答要素	書式	説明
RoutingSegmentID	文字列	1 つ以上のルート セグメント ID。
SegmentData	文字列	<p>ルート セグメントのすべての情報。これには、次の情報が含まれます。</p> <ul style="list-style-type: none"> <li>• PrimaryName</li> <li>• PrimaryNameLanguage</li> <li>• AlternateNameList</li> <li>• SegmentLength</li> <li>• SegmentLengthUnit</li> <li>• TimeTaken</li> <li>• TimeUnit</li> <li>• TurnAngle</li> <li>• TurnAngleUnit</li> <li>• speedOfTravel</li> <li>• speedOfTravelUnit</li> <li>• RoadType</li> <li>• SegmentDirection</li> <li>• StartJunctionType</li> <li>• IsRoundabout</li> <li>• IsTollRoad</li> <li>• PointsInSegment</li> </ul> <p>セグメントのルートジオメトリ(ポイント)は、PointsInSegment パラメータに返されます。</p>

## PersistentUpdate

PersistentUpdate によって、サーバー レベルでルーティング データの変更を行い、すべてのルート要求またはステージに適用できます。更新は、サーバーが再起動した後も維持されます。更新には、次の 4 タイプがあります。

1. ポイントの更新
2. セグメントの更新

## 3. 道路タイプの更新

## 4. リセットの更新

永続更新でこのような変更を行うことで、次の操作を実行できます。

- ポイントの除外
- セグメントの除外
- ポイント、セグメント、道路タイプの速度の設定
- ポイント、セグメント、道路タイプの速度の値による変更 (増加または減少)
- ポイント、セグメント、道路タイプの速度の割合による変更 (増加または減少)

**注：** PersistentUpdate は、サービス (Management Console および SOAP Web サービス) としてのみ利用可能です。PersistentUpdate を、ステージまたは REST API を介して利用することはできません。Java、C++、C、.NET、または COMAPI を介して利用することもできません。

**注：** 永続更新はルーティング データに対してシステム全体で行われる変更であり、すべての更新が永続するため、注意して使用する必要があります。

PersistentUpdate は、Spatial モジュールに含まれています。

## リソース URL

```
http://server:port/soap/PersistentUpdate
```

## 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:per="http://www.g1.com/services/PersistentUpdate"
xmlns:typ="http://www.g1.com/services/erm/types">
  <soapenv:Header/>
  <soapenv:Body>
    <per:PersistentUpdateRequest>
      <per:context>
        <per:account.id>admin</per:account.id>
        <per:account.password>admin</per:account.password>
      </per:context>
      <per:options>
        <per:DataSetResourceName>US</per:DataSetResourceName>
        <per:RestoreDefaults>N</per:RestoreDefaults>
      </per:options>
      <per:rows>
        <per:row>
          <per:PersistentUpdates>
```

```

        <typ:UpdateList>
          <typ:Update>
            <typ:PointUpdate>
              <typ:Point>
                <typ:Latitude>34.40691</typ:Latitude>
                <typ:Longitude>-80.062866</typ:Longitude>

                </typ:Point>
              <typ:SpeedUpdate>
                <typ:Velocity
VelocityUnit="mph">15</typ:Velocity>
                </typ:SpeedUpdate>
              </typ:PointUpdate>
            </typ:Update>
          </typ:UpdateList>
        </per:PersistentUpdates>
      </per:row>
    </per:rows>
  </per:PersistentUpdateRequest>
</soapenv:Body>
</soapenv:Envelope>

```

### 要求

オプションのパラメータ

入力オプション

表 96 : PersistentUpdate の入力オプション

パラメータ	説明
DataSetResourceName	更新プロセスで使用するデータが格納されたデータベースの名前。Spatial モジュールのルーティング データベース リソース ツールで指定されたデータベース名を使用してください。詳細については、『 <i>Spectrum™ Technology Platform 管理ガイド</i> 』を参照してください。
CoordinateSystem	入力される座標の座標系。  EPSG コードの詳細については、 <a href="http://www.spatialreference.org">www.spatialreference.org</a> を参照してください。EPSG に対してサポートされるコードスペースのリストを取得するには、SOAP リクエスト List Supported CoordSys by Code Space を Geometry サービス デモ ページ <code>http://&lt;server&gt;:&lt;port&gt;/Spatial/GeometryService/DemoPage.html</code> から送信します。

パラメータ	説明
RestoreDefaults	DataSetResourceName に対して行われたすべての永続更新が元の状態に戻されます。デフォルト値を復元するには Y を指定します。

## ポイント オプション

この優先設定セットを使用して、永続更新ごとにポイントの更新を設定することができます。ポイントの更新は、対応するポイント (緯度、経度) に適用される変更です。特定のポイントに対して、ポイントの除外、ポイントの速度の設定、値または割合によるポイントの速度の変更 (増加または減少) を行うことができます。

Point を Latitude と Longitude で指定してから、次のいずれかを指定する必要があります。要求ごとに 1 つ以上の更新を含めることができます。

**表 97 : ポイント更新オプション**

パラメータ	説明
ポイント	永続更新を実行するポイント。 <b>Latitude</b> と <b>Longitude</b> から構成されるポイントを指定する必要があります。
除外	指定されたポイントをすべてのルート計算から除外する文字列値です。ポイントを除外するには、ポイントを指定して、Y と定義した Exclude パラメータを含める必要があります。有効な値は Y (はい) と N (いいえ) です。
速度	速度単位と新しい速度を指定してポイントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。
SpeedIncrease	速度 (単位と値) または速度を増加させる割合を指定してポイントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。

パラメータ	説明
SpeedDecrease	速度 (単位と値) または速度を減少させる割合を指定してポイントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。

## セグメント オプション

この優先設定セットを使用して、永続更新ごとにセグメントの更新を設定することができます。セグメントの更新は、対応するルートセグメント ID に適用される変更です。特定のセグメントに対して、セグメントの除外、セグメントの速度の設定、値または割合によるセグメントの速度の変更 (増加または減少)、セグメントの道路タイプの変更を行うことができます。

`RoutingSegmentID` を指定してから、セグメントの更新のタイプを指定する必要があります。要求ごとに 1 つ以上の更新を含めることができます。

**表 98 : セグメント更新オプション**

パラメータ	説明
RoutingSegmentID	永続更新を実行するルートセグメント ID。
除外	指定されたセグメントをすべてのルート計算から除外する文字列値です。セグメントを除外するには、セグメント ID を指定して、Y と定義した <code>Exclude</code> パラメータを含める必要があります。有効な値は Y (はい) と N (いいえ) です。



パラメータ

説明

---

道路タイプ

## パラメータ

## 説明

ルート計算に使用するセグメントの道路タイプの値を変更する文字列値です。

RoadType には、次の種類があります。

- access way
- back road
- connector
- ferry
- footpath
- limited access dense urban
- limited access rural
- limited access suburban
- limited access urban
- local road dense urban
- local road rural
- local road suburban
- local road urban
- major local road dense urban
- major local road rural
- major local road suburban
- major local road urban
- major road dense urban
- major road rural
- major road suburban
- major road urban
- minor local road dense Urban
- minor local road rural
- minor local road suburban
- minor local road urban
- normal road dense urban
- normal road rural
- normal road rural
- normal road urban
- primary highway dense urban
- primary highway rural
- primary highway suburban
- primary highway urban
- ramp dense urban
- ramp limited access
- ramp major road
- ramp primary highway
- ramp rural
- ramp secondary highway
- ramp urban
- ramp suburban

## パラメータ

## 説明

---

	<ul style="list-style-type: none"> <li>• secondary highway dense urban</li> <li>• secondary highway rural</li> <li>• secondary highway suburban</li> <li>• secondary highway urban</li> </ul>
速度	速度単位と新しい速度を指定してセグメントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。
SpeedIncrease	速度 (単位と値) または速度を増加させる割合を指定してセグメントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。
SpeedDecrease	速度 (単位と値) または速度を減少させる割合を指定してセグメントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。

---

## 道路タイプ オプション

この優先設定セットを使用して、永続更新ごとに道路タイプの更新を設定することができます。道路タイプの更新は、対応する道路タイプに適用される変更です。特定の道路タイプに対して、道路タイプの速度の設定、値または割合による道路タイプの速度の変更 (増加または減少) を行うことができます。

RoadType を指定してから、更新のタイプを指定する必要があります。要求ごとに 1 つ以上の更新を含めることができます。

表 99 : 道路タイプ更新のオプション

パラメータ	説明
-------	----

---

道路タイプ	
-------	--

## パラメータ

## 説明

ルート計算に使用する道路タイプの速度の値を変更する文字列値です。

道路タイプには、次の種類があります。

- AccessWay
- Backroad
- Connector
- Ferry
- Footpath
- LimitedAccessDenseUrban
- LimitedAccessRural
- LimitedAccessSuburban
- LimitedAccessUrban
- LocalRoadDenseUrban
- LocalRoadRural
- LocalRoadSuburban
- LocalRoadUrban
- MajorLocalRoadDenseUrban
- MajorLocalRoadRural
- MajorLocalRoadSuburban
- MajorLocalRoadUrban
- MajorRoadDenseUrban
- MajorRoadRural
- MajorRoadSuburban
- MajorRoadUrban
- MinorLocalRoadDenseUrban
- MinorLocalRoadRural
- MinorLocalRoadSuburban
- MinorLocalRoadUrban
- NormalRoadDenseUrban
- NormalRoadRural
- NormalRoadSuburban
- NormalRoadUrban
- PrimaryHighwayDenseUrban
- PrimaryHighwayRural
- PrimaryHighwaySuburban
- PrimaryHighwayUrban
- RampDenseUrban
- RampLimitedAccess
- RampMajorRoad
- RampPrimaryHighway
- RampRural
- RampSecondaryHighway
- RampSuburban
- RampUrban
- SecondaryHighwayDenseUrban

パラメータ	説明
	<ul style="list-style-type: none"> <li>• SecondaryHighwayRural</li> <li>• SecondaryHighwaySuburban</li> <li>• SecondaryHighwayUrban</li> </ul>
速度	速度単位と新しい速度を指定してセグメントの新しい速度を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。
SpeedIncrease	速度 (単位と値) または速度を増加させる割合を指定してセグメントの速度の増加を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。
SpeedDecrease	速度 (単位と値) または速度を減少させる割合を指定してセグメントの速度の減少を定義する速度更新です。速度更新の場合、速度単位には <b>kph</b> (キロメートル/時)、 <b>mph</b> (マイル/時)、 <b>mps</b> (メートル/秒)、 <b>mtpm</b> (メートル/分) のいずれかの値を指定できます。

## リセット オプション

この優先設定セットを使用して、ポイント、セグメント、または道路タイプの更新ごとに更新をリセット (取り消し) することができます。この操作は、ポイント/セグメント/道路タイプに既に適用されている更新をサーバーからクリアして、デフォルト値に設定するだけです。

**表 100 : 更新のリセット オプション**

パラメータ	説明
PointReset	永続更新が行われたポイントをリセットします。Latitude と Longitude から構成されるポイントを指定してから、ポイントに対してリセットする更新のタイプを指定する必要があります。Reset Type のオプションは、除外と速度です。

パラメータ	説明
SegmentReset	永続更新が行われたセグメントをリセットします。RoutingSegmentIDを指定してから、そのセグメントに対してリセットする更新のタイプを指定する必要があります。Reset Type のオプションは、除外、道路タイプ、速度です。
RoadTypeReset	永続更新が行われた道路タイプをリセットします。道路タイプを指定すると、そのタイプに適用された速度の更新がリセットされます。

### PersistentUpdate の SOAP の例

使用可能な更新オプションをすべて使った標準的な PersistentUpdate の SOAP 要求を以下に示します(すべての要求構文を示すためのもので、実例ではありません)。複数の Update または Reset 定義を1つの PersistentUpdates に含めることができます。1つの更新タイプ(PointUpdate、SegmentUpdate、または RoadTypeUpdate)だけを1つの Update に含めることができます。同様に、1つのリセットタイプ(PointReset、SegmentReset、または RoadTypeReset)を1つの Reset に含めることができます。更新タイプまたはリセットタイプのいずれかに含めることができる更新またはリセットも1つだけです。

すべての更新に対してリセットを実行するには、DataSetResourceName を指定して、RestoreDefaults パラメータを Y に設定するだけです。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:per="http://www.g1.com/services/PersistentUpdate"
xmlns:typ="http://www.g1.com/services/erm/types">
  <soapenv:Header/>
  <soapenv:Body>
    <per:PersistentUpdateRequest>
      <per:options>
        <per:DataSetResourceName>US</per:DataSetResourceName>
        <per:CoordinateSystem>?</per:CoordinateSystem>
        <per:RestoreDefaults>N</per:RestoreDefaults>
      </per:options>
      <per:rows>
        <per:row>
          <per:PersistentUpdates>
            <typ:UpdateList>
              <typ:Update>
                <typ:PointUpdate>
                  <typ:Point>
                    <typ:Latitude>?</typ:Latitude>
                    <typ:Longitude>?</typ:Longitude>

```

```

        </typ:Point>
        <typ:SpeedUpdate>
          <typ:Velocity VelocityUnit=""/>
          <typ:SpeedIncrease>
            <typ:Velocity VelocityUnit=""/>
            <typ:Percentage?></typ:Percentage>
          </typ:SpeedIncrease>
          <typ:SpeedDecrease>
            <typ:Velocity
VelocityUnit=""?></typ:Velocity>
              <typ:Percentage?></typ:Percentage>
            </typ:SpeedDecrease>
          </typ:SpeedUpdate>
          <typ:Exclude?></typ:Exclude>
        </typ:PointUpdate>
        <typ:SegmentUpdate>
          <typ:RoutingSegmentID?></typ:RoutingSegmentID>
          <typ:SpeedUpdate>
            <typ:Velocity
VelocityUnit=""?></typ:Velocity>
              <typ:SpeedIncrease>
                <typ:Velocity
VelocityUnit=""?></typ:Velocity>
                  <typ:Percentage?></typ:Percentage>
                </typ:SpeedIncrease>
                <typ:SpeedDecrease>
                  <typ:Velocity
VelocityUnit=""?></typ:Velocity>
                    <typ:Percentage?></typ:Percentage>
                </typ:SpeedDecrease>
              </typ:SpeedUpdate>
              <typ:RoadType?></typ:RoadType>
              <typ:Exclude?></typ:Exclude>
            </typ:SegmentUpdate>
            <typ:RoadTypeUpdate>
              <typ:RoadType?></typ:RoadType>
              <typ:SpeedUpdate>
                <typ:Velocity
VelocityUnit=""?></typ:Velocity>
                  <typ:SpeedIncrease>
                    <typ:Velocity
VelocityUnit=""?></typ:Velocity>
                      <typ:Percentage?></typ:Percentage>
                    </typ:SpeedIncrease>
                    <typ:SpeedDecrease>
                      <typ:Velocity
VelocityUnit=""?></typ:Velocity>
                        <typ:Percentage?></typ:Percentage>
                    </typ:SpeedDecrease>
                  </typ:SpeedUpdate>
                </typ:RoadTypeUpdate>
              </typ:Update>

```



```

</typ:UpdateList>
<typ:ResetList>
  <typ:Reset>
    <typ:PointReset ResetType="?">
      <typ:Point>
        <typ:Latitude?></typ:Latitude>
        <typ:Longitude?></typ:Longitude>
      </typ:Point>
    </typ:PointReset>
    <typ:SegmentReset ResetType="?">
      <typ:RoutingSegmentID?></typ:RoutingSegmentID>
      </typ:SegmentReset>
      <typ:RoadTypeReset>
        <typ:RoadType?></typ:RoadType>
      </typ:RoadTypeReset>
    </typ:Reset>
  </typ:ResetList>
</per:PersistentUpdates>
</per:row>
</per:rows>
</per:PersistentUpdateRequest>
</soapenv:Body>
</soapenv:Envelope>

```

**応答****応答**

PersistentUpdate は次のフィールドを返します。

**表 101 : PersistentUpdate の出力**

応答要素	書式	説明
SuccessMessage	文字列	更新またはリセットが正常に完了した場合に返されます。

**Universal Addressing モジュール****AutoCompleteLoqate**

AutoCompleteLoqate は、住所データのリアルタイム入力に対して高速で正確な結果を返します。フォームに入力された各文字に基づいて瞬時に結果が返され、正確なデータのみがデータベース

に入力されることを保証します。**AutoCompleteLoqate** には [高度な検索を優先] オプションも用意されており、インデックス ファイル形式のデータを使用することによって、238 カ国に対して入力時間を最大 80% 短縮します。

### リソース URL

```
http://server:port/soap/AutoCompleteLoqate
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aut="http://www.pb.com/spectrum/services/AutoCompleteLoqate"
xmlns:spec="http://spectrum.pb.com/"
  <soapenv:Header/>
  <soapenv:Body>
    <aut:AutoCompleteLoqateRequest>
      <aut:input_port>
        <aut:Address>
          <aut:AddressLine1>1 Global</aut:AddressLine1>
        </aut:Address>
      </aut:input_port>
    </aut:AutoCompleteLoqateRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

注: この例では、内容を理解しやすいように空の応答要素を削除し、最初の3つの住所マッチのみを示しています。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
    <ns3:AutoCompleteLoqateResponse xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/AutoCompleteLoqate">
      <ns3:output_port>
        <ns3:Address>
          <ns3:ProcessedBy>LOQATE</ns3:ProcessedBy>
          <ns3:HouseNumber>1</ns3:HouseNumber>
          <ns3:AddressLine1>1 Global Vw</ns3:AddressLine1>
          <ns3:FirmName>Map Info</ns3:FirmName>
          <ns3:City>Troy</ns3:City>
          <ns3:StateProvince>NY</ns3:StateProvince>
          <ns3:PostalCode>12180-8399</ns3:PostalCode>
          <ns3:PostalCode.AddOn>8399</ns3:PostalCode.AddOn>
          <ns3:Country>United States</ns3:Country>
        </ns3:Address>
```

```

    <ns3:Address>
      <ns3:ProcessedBy>LOQATE</ns3:ProcessedBy>
      <ns3:HouseNumber>1</ns3:HouseNumber>
      <ns3:AddressLine1>1 Global Pl</ns3:AddressLine1>
      <ns3:City>Glendale</ns3:City>
      <ns3:StateProvince>AZ</ns3:StateProvince>
      <ns3:PostalCode>85306-3216</ns3:PostalCode>
      <ns3:PostalCode.AddOn>3216</ns3:PostalCode.AddOn>
      <ns3:Country>United States</ns3:Country>
    </ns3:Address>
    <ns3:Address>
      <ns3:ProcessedBy>LOQATE</ns3:ProcessedBy>
      <ns3:HouseNumber>1</ns3:HouseNumber>
      <ns3:AddressLine1>1 Global Dr</ns3:AddressLine1>
      <ns3:City>Olive Hill</ns3:City>
      <ns3:StateProvince>KY</ns3:StateProvince>
      <ns3:PostalCode>41164-6739</ns3:PostalCode>
      <ns3:PostalCode.AddOn>6739</ns3:PostalCode.AddOn>
      <ns3:Country>United States</ns3:Country>
    </ns3:Address>
  </ns3:output_port>
</ns3:AutoCompleteLoqateResponse>
</soap:Body>
</soap:Envelope>

```

**要求**

入力データのパラメータ

AutoCompleteLoqate の入力の一覧を以下の表に示します。

**表 102 : 入力フォーマット**

パラメータ	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。
AddressLine4	4 行目の住所行。

パラメータ	説明
City	都市名。
Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2 桁の ISO 国コード</li> <li>• 3 桁の UPU 国コード</li> <li>• 英語の国名</li> </ul> <p>ISO コードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a> (735ページ) を参照してください。</p>
FirmName	会社名または企業名。
PostalCode	住所の郵便番号。
StateProvince	州または省。

### オプションのパラメータ

表 103 : AutoCompleteLoqate のオプション

パラメータ	説明
Database.Loqate	住所処理に使用するデータベースを指定します。Management Console の [データベース リソース] パネルで定義されたデータベースのみが使用可能です。
OutputCasing	<p>出力データの大文字と小文字の区別を指定します。次のいずれかです。</p> <p><b>M</b> 出力に大文字と小文字を混在させます (デフォルト)。例:</p> <p>123 Main StMytown FL 12345</p> <p><b>U</b> 出力に大文字を使用します。例:</p> <p>123 MAIN STMYTOWN FL 12345</p>

## パラメータ

## 説明

## HomeCountry

デフォルトの国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がドイツにある場合は、ドイツを指定します。有効な国名には次のものがあります。

Afghanistan, Albania, Algeria, American Samoa, 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, Equatorial 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, Suriname, 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

## OutputCountryFormat

**Country** 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。

- E** 英語の国名を使用します (デフォルト)。
- I** 国名の代わりにその国の 2 文字の ISO の略語を使用します。
- U** 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。

## パラメータ

## 説明

---

**OutputScript** 出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。

**Input** 書き直しを実行せず、入力と同じスクリプトで出力します(デフォルト)。

**Native** 使用可能な場合は、選択した国のネイティブ スクリプトで出力します。

**Latn** 英語の値を使用します。

---

**MaximumResults**

**AutoCompleteLoqate** が返す住所の最大数。デフォルトは 10 です。

---

**isPowersearchEnable**

インデックス ファイル形式のデータを使用することによって、240 カ国に対して入力時間を最大 80% 短縮します。検索を実行すると、Loqate エンジンはず、該当するインデックスを検索します。インデックスが存在する場合は、候補住所のリストを直ちに返そうと試みます。インデックスが存在しないか、インデックスによって結果が1つも返されない場合は、オリジナルの検索処理を開始します。

注：高度な検索は、入力ファイルにフィールドが 2つだけ存在する場合に実行可能です。1つは **Country** フィールドで、もう 1つはいずれかの **AddressLine** フィールドです。このオプションを選択し、入力ファイルにそれ以外のフィールドが含まれる場合は、オリジナルの検索処理が自動的に開始されます。

検索を行うために、**Auto Complete** インデックスは米国内の検索に対しては最初の 10 文字まで、その他すべての対象国内の検索に対しては最初の 15 文字までを使用します。空白と句読文字は、この文字数にカウントされません。

高度な検索は、ボツワナ、エチオピア、インド、カザフスタン、マレーシア、モンゴル、セントクリストファー・ネイビス、およびサンマリノでは使用できません。

注：高度な検索を使用するには、有効なライセンスが必要です。高度な検索のライセンスを取得していないか、ライセンスの期限が切れている状態でこのオプションを選択すると、エラーとなります。

---

パラメータ	説明
IsDuplicateHandlingMaskEnable	<p>重複処理マスクを有効にし、重複レコードの処理および削除の方法を指定します。次のオプションから 1 つ以上を選択します。</p> <p><b>S</b> デフォルトで選択されています。入力の前処理により、単一フィールドで発生している重複を削除します。</p> <p><b>C</b> デフォルトで選択されています。入力の前処理により、すべてのフィールドわたり重複を削除します。</p> <p><b>T</b> 入力の前処理により、標準住所フィールドでないフィールド内の重複を削除します。</p> <p><b>F</b> デフォルトで選択されています。検証の出力の後処理により、検証されていないフィールドから重複を削除します。</p>
FailJobOnDataLicenseError	<p>データ ライセンス エラーの発生時に Spectrum Technology Platform がどのように応答するかを指定します。</p> <p>ジョブのエラー      データライセンスエラーが発生した場合、ジョブ全体をエラーにします。</p> <p>レコードのエラー      データライセンスエラーの発生原因となったレコードをエラーにし、処理を続行します。</p>

## 応答

AutoCompleteLoqate の出力はオプションであり、[AutoCompleteLoqate オプション] ダイアログボックスの [出力フィールド] セクションで選択したフィールドに直接対応します。

**表 104 : AutoCompleteLoqate の出力**

応答要素	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。

応答要素	説明
AddressLine4	4 行目の住所行。
City	都市名。
Country	3 文字の ISO 3166-1 Alpha 3 国コード。ISO コードの一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。
FirmName	企業名。
HouseNumber	候補住所の家番号が含まれる範囲の終了家番号。
PostalCode	郵便番号。
PostalCode.AddOn	ZIP + 4 <sup>®</sup> Code の末尾 4 桁。
ProcessedBy	住所を処理した住所コーダーを示します。 <b>LOQATE</b> Loqate コーダーが住所を処理しました。
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。 <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• RequestFailed</li> <li>• NoLookupAddressFound</li> </ul>



## 応答要素

## 説明

## Status.Description

問題の説明 (ある場合)。

**Did not return multiples**

入力住所はデータベース内の1つの住所とのみマッチしました。AutoCompleteLoqate は、一致する可能性のある住所が複数見つかった場合のみデータを返します。

**Not able to look up the address pattern**

AutoCompleteLoqate は、部分的な住所を処理できません。

**AutoCompleteLoqate のサンプル Web アプリケーション**

Auto Complete Loqate 機能を示すサンプル Web アプリケーションにアクセスできます。部分的な住所を入力すると、このアプリケーションは Auto Complete Loqate REST Web サービスを呼び出します。この Web サービスは提案される住所を返します。

注：この機能を使用する前に、Management Console で Auto Complete Loqate データベースリソースを追加し、そのデータベースリソースを Auto Complete Loqate Service に保存する必要があります。

1. Spectrum™ Technology Platform サーバーが実行されていることを確認します。
2. Web ブラウザを開いて、`http://<servername>:<port>/autocomplete` に移動します。たとえば、サーバーの名前が "myserver" であり、既定の HTTP ポート 8080 を使う場合は、`http://myserver:8080/autocomplete` に移動します。

注：このサイトは、Internet Explorer 8.0 またはそれ以降、Chrome、Mozilla Firefox で最も適切に表示されます。

3. ログイン画面が表示されたら、ユーザ名として "guest" を入力し、パスワードフィールドは空白のままにします。
4. **[OK]** をクリックします。
5. ドロップダウン リストから国を選択します。
6. 提供されている任意のフィールドで住所の入力を開始します。
7. 提案された住所のリストから選択します。
8. 新しい呼び出しを開始するには、**[リセット]** をクリックして、前回の呼び出しで使用したフィールドをクリアします。

## GetCandidateAddresses

**GetCandidateAddresses** は、与えられた入力住所にマッチするとみなされる住所のリストを返します。**GetCandidateAddresses** は、入力住所が郵便データベースの複数の住所にマッチする場合のみ、候補の住所を返します。入力住所が、郵便データベースの 1 つの住所のみにマッチする場合は、住所データを返しません。

米国およびカナダ以外の住所については、**ValidateAddress** が返す複数のマッチ結果と、**GetCandidateAddresses** が同じ住所に対して返す結果の間に、矛盾が存在する場合があります。矛盾した結果が得られるのはおそらく、**ValidateAddress** のパフォーマンス チューニング設定で 100 以外の値を設定しているためです。**GetCandidateAddresses** と **ValidateAddress** で矛盾のない結果を得るには、パフォーマンス チューニング オプションを 100 に設定します。

注：デフォルトでは、**GetCandidateAddresses** は個々の家番号との一致は確認しません。各通りの家番号の範囲とのマッチングを行います。**GetCandidateAddresses** は通り名、都市名、州/省名、および郵便番号を特定した後、入力された家番号が、マッチした通り名の家番号の範囲に含まれるかどうかを確認します。ユニット番号についても同様の処理が行われます。個々の家番号が有効であることを確認するには、**ValidateAddress Delivery Point Validation (DPV)** 処理オプションを使用する必要があります。DPV 処理は、米国住所に対してのみ使用可能です。

カナダのコーダーには、特定の郵便番号を入力として受け取り、その郵便番号のデータベースに格納された通り情報を返す逆検索ルーチンが含まれています。この機能を使用するには、**PostalCode** フィールドにカナダの郵便番号を入力します。カナダの郵便番号を入力した場合の結果については、以下の 2 つめの例を参照してください。

**GetCandidateAddresses** は、**Universal Addressing** モジュールに含まれています。

### リソース URL

```
http://server:port/soap/GetCandidateAddresses
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetCandidateAddresses"
xmlns:spec="http://spectrum.pb.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetCandidateAddressesRequest>
      <get:input_port>
        <get:Address>
          <get:AddressLine1>P.O. Box 1</get:AddressLine1>
```

```

        <get:City>New York</get:City>
        <get:StateProvince>NY</get:StateProvince>
    </get:Address>
</get:input_port>
</get:GetCandidateAddressesRequest>
</soapenv:Body>
</soapenv:Envelope>

```

応答は次のようになります。

注：この例では、空の応答要素を削除しています。最初の候補住所を 2 つだけ示します。

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:GetCandidateAddressesResponse
xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/GetCandidateAddresses">
      <ns3:output_port>
        <ns3:Address>
          <ns3:ProcessedBy>USA</ns3:ProcessedBy>
          <ns3:RecordType>PostOfficeBox</ns3:RecordType>
          <ns3:MatchLevel>A</ns3:MatchLevel>
          <ns3:AddressLine1>PO Box 1</ns3:AddressLine1>
          <ns3:HouseNumberLow>1</ns3:HouseNumberLow>
          <ns3:HouseNumberHigh>60</ns3:HouseNumberHigh>
          <ns3:HouseNumberParity>B</ns3:HouseNumberParity>
          <ns3:City>New York</ns3:City>
          <ns3:StateProvince>NY</ns3:StateProvince>
          <ns3:PostalCode>10002</ns3:PostalCode>
          <ns3:PostalCode.AddOn>0001</ns3:PostalCode.AddOn>
          <ns3:Country>USA</ns3:Country>
        </ns3:Address>
        <ns3:Address>
          <ns3:ProcessedBy>USA</ns3:ProcessedBy>
          <ns3:RecordType>PostOfficeBox</ns3:RecordType>
          <ns3:MatchLevel>A</ns3:MatchLevel>
          <ns3:AddressLine1>PO Box 1</ns3:AddressLine1>
          <ns3:HouseNumberLow>1</ns3:HouseNumberLow>
          <ns3:HouseNumberHigh>9</ns3:HouseNumberHigh>
          <ns3:HouseNumberParity>B</ns3:HouseNumberParity>
          <ns3:City>New York</ns3:City>
          <ns3:StateProvince>NY</ns3:StateProvince>
          <ns3:PostalCode>10008</ns3:PostalCode>
          <ns3:PostalCode.AddOn>0001</ns3:PostalCode.AddOn>
          <ns3:Country>USA</ns3:Country>
        </ns3:Address>
      </ns3:output_port>
    </ns3:GetCandidateAddressesResponse>
  </soap:Body>
</soap:Envelope>

```

**要求**

## 入力データのパラメータ

GetCandidateAddresses の入力の一覧を以下の表に示します。

**表 105 : 入力フォーマット**

パラメータ	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。 米国またはカナダの住所には適用されません。
AddressLine4	4 行目の住所行。 米国またはカナダの住所には適用されません。
AddressLine5	5 行目の住所行。 英国住所にのみ適用されます。通り名、ユニット番号、ビルディング番号などを 含めることができます。
City	都市名。
StateProvince	州または省。 米国の住所に限り、州を StateProvince フィールドの代わりに City フィールドに 入れることができます。

パラメータ	説明
PostalCode	<p>住所の郵便番号。米国住所の場合は、次のいずれかの形式の ZIP Code™ になります。</p> <p>9999999999-9999A9A9A9A9A 9A99999 999</p> <p>注：カナダ住所の場合は、このフィールドのみを入力すると、候補となる住所データが返されます。その他の国の場合は、AddressLine1 と AddressLine2 も入力する必要があります。</p>
Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"><li>• 2 桁の ISO 国コード</li><li>• 3 桁の UPU 国コード</li><li>• 英語の国名</li><li>• フランス語の国名</li><li>• ドイツ語の国名</li><li>• スペイン語の国名</li></ul> <p>ISO コードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a>（735ページ）を参照してください。</p>
FirmName	会社名または企業名。
USUrbanName	米国住所都市化名。主にプエルトリコの住所に使用します。

## オプションのパラメータ

表 106 : GetCandidateAddresses オプション

パラメータ	説明
PerformUSProcessing	<p>米国住所を処理するかどうかを指定します。米国住所処理を有効にすると、GetCandidateAddresses は、米国住所の候補となる住所の取得を試みます。米国住所処理を無効にすると、米国住所は失敗します。つまり、Status 出力フィールドに "F" が設定されて返されます。出力フィールド Status.Code は、"DisabledCoder" となります。米国住所処理のライセンスを取得していない場合は、ジョブに米国住所が含まれるか否かにかかわらず、米国住所処理を無効にしなければ、ジョブを正常に実行することはできません。</p> <p>注：米国住所を正常に処理するには、米国住所処理の有効なライセンスを取得する必要があります。米国住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、米国住所処理を有効にすると、エラーとなります。</p> <p><b>Y</b>      はい、米国住所を処理します (デフォルト)。</p> <p><b>N</b>      いいえ、米国住所を処理しません。</p>
Database.US	<p>米国住所処理に使用するデータベースを指定します。Management Console の <b>【米国データベース リソース】</b> パネルで定義されたデータベースのみが使用可能です。</p>

## パラメータ

## 説明

---

**PerformCanadianProcessing**

カナダ住所を処理するかどうかを指定します。カナダ住所処理を有効にすると、**GetCandidateAddresses** は、カナダ住所の候補となる住所の取得を試みます。カナダ住所処理を無効にした場合、**Status** フィールドに "F" が設定されカナダ住所は失敗します。出力フィールド **Status.Code** は、"DisabledCoder" となります。カナダ住所処理のライセンスを取得していない場合は、ジョブにカナダ住所が含まれるか否かにかかわらず、カナダ住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：カナダ住所を正常に処理するには、カナダ住所処理の有効なライセンスを取得する必要があります。カナダ住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、カナダ住所処理を有効にすると、エラーとなります。

**Y** カナダ住所を処理します (デフォルト)。

**N** カナダ住所を処理しません。

---

**Database.Canada**

カナダ住所処理に使用するデータベースを指定します。**Management Console** の **[カナダ データベース リソース]** パネルで定義されたデータベースのみが使用可能です。

---

**PerformInternationalProcessing**

国際住所 (米国およびカナダ以外の住所) を処理するかどうかを指定します。国際住所処理を有効にすると、**GetCandidateAddresses** は、国際住所の候補となる住所の取得を試みます。国際住所処理を無効にした場合、**Status** フィールドに "F" が設定され国際住所は失敗します。出力フィールド **Status.Code** は、"DisabledCoder" となります。国際住所処理のライセンスを取得していない場合は、ジョブに国際住所が含まれるか否かにかかわらず、国際住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：国際住所を正常に処理するには、国際住所処理の有効なライセンスを取得する必要があります。国際住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、国際住所処理を有効にすると、エラーとなります。

**Y** 国際住所を処理します (デフォルト)。

**N** 国際住所を処理しません。

パラメータ	説明
Database.International	国際的な住所の検証に使用するデータベースを指定します。 Management Console の【国際データベース リソース】パネルで定義されたデータベースのみが使用可能です。
OutputCasing	出力データの大文字と小文字の区別を指定します。次のいずれかです。 <b>M</b> 出力に大文字と小文字を混在させます (デフォルト)。例: 123 Main StMytown FL 12345 <b>U</b> 出力に大文字を使用します。例: 123 MAIN STMYTOWN FL 12345
MaximumResults	GetCandidateAddresses が返す候補住所の最大数。デフォルトは 10 です。最大数は 10 になります。
OutputShortCityName	米国住所に対しては、USPS® が承認する都市の略称がある場合に、それを返すかどうかを指定します。USPS® は、14 文字以上の都市名に対し、略称を定めています。都市の略称は 13 文字以下で、宛名ラベルのサイズが限られている場合に使用できます。短い都市名が存在しない都市に対しては、正式な都市名が返されます。 <b>Y</b> 短い都市名を返します。 <b>N</b> 短い都市名を返しません。



## パラメータ

## 説明

## DualAddressLogic

(米国住所のみ)。ストリート情報と PO Box/地方配送路/Highway Contract 情報の両方が住所に含まれる場合に、GetCandidateAddresses が、ストリート一致を返すか、または PO Box/地方配送路/Highway Contract 一致を返すかを制御します。詳細については、[二重住所ロジックについて](#) (588ページ) を参照してください。

- N** (デフォルト)USPS®CASS™ の規則では、以下の優先順位に基づいて、返す住所を決定します。
1. PO Box
  2. Firm
  3. Highrise
  4. Street
  5. Rural Route
  6. General Delivery
- S** 住所行に関係なく、通り一致を返します。
- P** 住所行に関係なく、PO Box 一致を返します。

## StreetMatchingStrictness

通り名のマッチングの精度 (米国住所のみ)。

- E** 入力された通り名は、データベースに完全に一致する必要があります。
- T** マッチングアルゴリズムは "厳格" です。
- M** マッチングアルゴリズムは "中" です (デフォルト)。
- L** マッチングアルゴリズムは "あいまい" です。

## FirmMatchingStrictness

企業名マッチングの精度 (米国住所のみ)。

- E** 入力された企業名は、データベースに完全に一致する必要があります。
- T** マッチングアルゴリズムは "厳格" です。
- M** マッチングアルゴリズムは "中" です (デフォルト)。
- L** マッチングアルゴリズムは "あいまい" です。

## パラメータ

## 説明

## DirectionalMatchingStrictness

道順マッチングの精度。

- E** 入力された道順は、データベースに完全に一致する必要があります。
- T** マッチングアルゴリズムは "厳格" です。
- M** マッチングアルゴリズムは "中" です (デフォルト)。
- L** マッチングアルゴリズムは "あいまい" です。

## PerformESM

Enhanced Street Matching (ESM) を実行するかどうかを指定します。ESM は、通常の住所検証プロセスでマッチしなかった任意の入力住所に対して、追加データによる別のマッチングロジックを適用します。ESM は、米国住所にのみ適用されます。

- Y** ESM 処理を実行します。
- N** ESM 処理を実行しません (デフォルト)。

## AddressLineSearchOnFail

ValidateAddress において、住所行で都市、州/省、郵便番号の検索を行うかどうかを指定します。

このオプションにより、ValidateAddress において、City、StateProvince、および PostalCode の各入力フィールドの値を使用して住所にマッチする結果が得られなかった場合に、AddressLine 入力フィールドで都市、州/省、郵便番号、および国を検索することができます。

入力住所において、AddressLine フィールドに都市、州/省、および郵便番号の情報が存在する場合は、このオプションを有効にすることを検討してください。

入力住所において、City、State/Province、および PostalCode フィールドが使用されている場合は、このオプションを無効にしてください。このオプションを有効にしてこれらのフィールドを使用すると、ValidateAddress がこれらのフィールド値の修正 (例えば、スペルミスのある都市名など) に失敗する可能性が高くなります。

- Y** AddressLine フィールドを検索します (デフォルト)。
- N** いいえ、AddressLine フィールドを検索しません。

## 応答

GetCandidateAddresses は、次の出力を返します。

表 107 : GetCandidateAddresses の出力

応答要素	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。
AddressLine4	4 行目の住所行。
AddressLine5	英国住所にのみ適用されます。住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 5 行目です。住所の妥当性が確認できなかった場合は、入力住所の 5 行目がそのまま出力されます。
CanadianDeliveryInstallation AreaName	配達施設名 (カナダ住所のみ)
CanadianDeliveryInstallation QualifierName	配達施設の修飾子 (カナダ住所のみ)
CanadianDeliveryInstallation Type	配達施設の種類 (カナダ住所のみ)
City	都市名。
Country	3 文字の ISO 3166-1 Alpha 3 国コード。ISO コードの一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735 ページ) を参照してください。
FirmName	企業名。

応答要素	説明
HouseNumberHigh	候補住所の家番号が含まれる範囲の終了家番号。
HouseNumberLow	候補住所の家番号が含まれる範囲の開始家番号。
HouseNumberParity	HouseNumberLow と HouseNumberHigh の間の家番号の番号付けスキームを次のように示します。 <ul style="list-style-type: none"> <li><b>E</b> 偶数値のみ</li> <li><b>O</b> 奇数値のみ</li> <li><b>B</b> 両方</li> </ul>
MatchLevel	米国およびカナダ以外の住所に対し、候補住所のマッチ レベルを示します。米国およびカナダの住所では常に "A" になります。次のいずれかです。 <ul style="list-style-type: none"> <li><b>A</b> 候補は通りレベルで入力住所にマッチします。</li> <li><b>B</b> 候補は州/省レベルで入力住所にマッチします。</li> </ul>
PostalCode	郵便番号。米国では、ZIP Code™になります。
PostalCode.AddOn	ZIP + 4® Code の末尾 4 桁。米国住所にのみ適用されます。
RecordType	米国およびカナダの郵政当局によって定義されている住所レコードのタイプ(米国およびカナダの住所のみサポート): <ul style="list-style-type: none"> <li>• FirmRecord</li> <li>• GeneralDelivery</li> <li>• HighRise</li> <li>• PostOfficeBox</li> <li>• RRHighwayContract</li> <li>• Normal</li> </ul>
RecordType.Default	"デフォルト" マッチを示すコード <ul style="list-style-type: none"> <li><b>Y</b> 住所はデフォルト レコードにマッチしています。</li> <li><b>NULL</b> 住所はデフォルト レコードにマッチしていません。</li> </ul>

応答要素	説明										
StateProvince	州または省の省略形。										
Status	<p>マッチの成功または失敗。</p> <table> <tr> <td><b>NULL</b></td> <td>成功</td> </tr> <tr> <td><b>F</b></td> <td>失敗</td> </tr> </table>	<b>NULL</b>	成功	<b>F</b>	失敗						
<b>NULL</b>	成功										
<b>F</b>	失敗										
Status.Code	<p>失敗の原因 (ある場合)。次のいずれかの値になります。</p> <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• RequestFailed</li> </ul>										
Status.Description	<p>問題の説明 (ある場合)。</p> <table> <tr> <td><b>Did not return multiples</b></td> <td>入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddresses は、一致する可能性のある住所が複数見つかった場合にデータを返します。</td> </tr> <tr> <td><b>Number of candidates is not greater than 1</b></td> <td>入力住所はデータベース内の複数の住所にマッチしましたが、住所が返されませんでした。</td> </tr> <tr> <td><b>PerformUSProcessing disabled</b></td> <td>Status.Code=DisabledCoder の場合にこの値が表示されます。</td> </tr> <tr> <td><b>PerformCanadianProcessing disabled</b></td> <td>Status.Code=DisabledCoder の場合にこの値が表示されます。</td> </tr> <tr> <td><b>PerformInternationalProcessing disabled</b></td> <td>Status.Code=DisabledCoder の場合にこの値が表示されます。</td> </tr> </table>	<b>Did not return multiples</b>	入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddresses は、一致する可能性のある住所が複数見つかった場合にデータを返します。	<b>Number of candidates is not greater than 1</b>	入力住所はデータベース内の複数の住所にマッチしましたが、住所が返されませんでした。	<b>PerformUSProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。	<b>PerformCanadianProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。	<b>PerformInternationalProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。
<b>Did not return multiples</b>	入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddresses は、一致する可能性のある住所が複数見つかった場合にデータを返します。										
<b>Number of candidates is not greater than 1</b>	入力住所はデータベース内の複数の住所にマッチしましたが、住所が返されませんでした。										
<b>PerformUSProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。										
<b>PerformCanadianProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。										
<b>PerformInternationalProcessing disabled</b>	Status.Code=DisabledCoder の場合にこの値が表示されます。										
UnitNumberHigh	候補住所のユニット番号が含まれる範囲の終了ユニット番号。										
UnitNumberLow	候補住所のユニット番号が含まれる範囲の開始ユニット番号。										

応答要素	説明
UnitNumberParity	UnitNumberLow と UnitNumberHigh の間のユニット番号の番号付けスキームを次のように示します。 <b>E</b> 偶数値のみ <b>O</b> 奇数値のみ <b>B</b> 両方
USUrbanName	妥当性が確認された都市の都市化名。都市化名は、主にプエルトリコ住所に使用されます。

### GetCandidateAddressesLoqate

GetCandidateAddressesLoqate は、与えられた入力住所に一致するとみなされる住所のリストを返します。GetCandidateAddressesLoqate は、入力住所が郵便データベースの複数の住所にマッチする場合のみ、候補の住所を返します。入力住所が、郵便データベースの1つの住所のみにマッチする場合は、住所データを返しません。[Country] 入力フィールドは必須です。このフィールドが空白の場合、出力は返されません。

注：デフォルトでは、GetCandidateAddressesLoqate は個々の家番号との一致は確認しません。各通りの家番号の範囲とのマッチングを行います。GetCandidateAddressesLoqate は通り名、都市名、州/省名、および郵便番号を特定した後、入力された家番号が、マッチした通り名の家番号の範囲に含まれるかどうかを確認します。ユニット番号についても同様の処理が行われます。

GetCandidateAddressesLoqate は、Universal Addressing モジュールに含まれています。

### リソース URL

```
http://server:port/soap/GetCandidateAddressesLoqate
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetCandidateAddressesLoqate"
xmlns:spec="http://spectrum.pb.com/">
  <soapenv:Header/>
```

```

<soapenv:Body>
  <get:GetCandidateAddressesLoqateRequest>
    <get:input_port>
      <get:Address>
        <get:AddressLine1>PO Box 1</get:AddressLine1>
        <get:City>New York</get:City>
        <get:StateProvince>NY</get:StateProvince>
      </get:Address>
    </get:input_port>
  </get:GetCandidateAddressesLoqateRequest>
</soapenv:Body>
</soapenv:Envelope>

```

応答は次のようになります。

注：この例では、空の応答要素を削除しています。最初の候補住所を 2 つだけ示します。

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:GetCandidateAddressesLoqateResponse
      xmlns:ns2="http://spectrum.pb.com/"
      xmlns:ns3="http://www.pb.com/spectrum/services/GetCandidateAddressesLoqate">
      <ns3:output_port>
        <ns3:Address>
          <ns3:ProcessedBy>LOQATE</ns3:ProcessedBy>
          <ns3:AddressLine1>PO Box 101</ns3:AddressLine1>
          <ns3:City>New York Mls</ns3:City>
          <ns3:StateProvince>NY</ns3:StateProvince>
          <ns3:PostalCode>13417-0101</ns3:PostalCode>
          <ns3:PostalCode.AddOn>0101</ns3:PostalCode.AddOn>
          <ns3:Country>USA</ns3:Country>
        </ns3:Address>
        <ns3:Address>
          <ns3:ProcessedBy>LOQATE</ns3:ProcessedBy>
          <ns3:AddressLine1>PO Box 102</ns3:AddressLine1>
          <ns3:City>New York Mls</ns3:City>
          <ns3:StateProvince>NY</ns3:StateProvince>
          <ns3:PostalCode>13417-0102</ns3:PostalCode>
          <ns3:PostalCode.AddOn>0102</ns3:PostalCode.AddOn>
          <ns3:Country>USA</ns3:Country>
        </ns3:Address>
      </ns3:output_port>
    </ns3:GetCandidateAddressesLoqateResponse>
  </soap:Body>
</soap:Envelope>

```

**要求**

## 入力データのパラメータ

GetCandidateAddressesLoqate の入力の一覧を以下の表に示します。

**表 108 : 入力フォーマット**

パラメータ	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。
AddressLine4	4 行目の住所行。
City	都市名。
Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2 桁の ISO 国コード</li> <li>• 3 桁の UPU 国コード</li> <li>• 英語の国名</li> </ul> <p>ISO コードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a> (735ページ) を参照してください。</p> <p>注：このフィールドは必須です。このフィールドが空白の場合、出力は返されません。</p>
FirmName	会社名または企業名。
PostalCode	住所の郵便番号。米国住所の場合は、次のいずれかの形式の ZIP Code™ になります。



パラメータ	説明
StateProvince	州または省。 米国住所に限り、州は、StateProvince フィールドではなく、City フィールドに入力することもできます。

## オプションのパラメータ

表 109 : GetCandidateAddressesLoqate のオプション

パラメータ	説明
Database.Loqate	住所処理に使用するデータベースを指定します。Management Console で定義されたデータベースのみが使用可能です。
OutputCasing	出力データの大文字と小文字の区別を指定します。次のいずれかです。 <ul style="list-style-type: none"> <li><b>M</b> 出力に大文字と小文字を混在させます (デフォルト)。例: 123 Main StMytown FL 12345</li> <li><b>U</b> 出力に大文字を使用します。例: 123 MAIN STMYTOWN FL 12345</li> </ul>
CandidateProcessOption	候補を検索する方法を指定します。次のいずれかです。 <ul style="list-style-type: none"> <li><b>S</b> 住所の全体または一部を入力し、近似一致結果のリストを出力として返します (デフォルト)。</li> <li><b>V</b> 住所行、住所コンポーネント、またはその両方を組み合わせて住所情報を入力し、入力により近く一致する結果を出力として返します。</li> </ul>

## パラメータ

## 説明

## HomeCountry

デフォルトの国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がドイツにある場合は、ドイツを指定します。**GetCandidateAddressLoqate** は、[StateProvince]、[PostalCode]、および [Country] の各住所フィールドから国を特定できなかった場合、指定された国を使用して、住所の検証を試みます。有効な国名には次のものがあります。

Afghanistan, Albania, Algeria, American Samoa, 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, Equatorial 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, Suriname, 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

## OutputCountryFormat

**Country** 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。

- E** 英語の国名を使用します (デフォルト)。
- I** 国名の代わりにその国の 2 文字の ISO の略語を使用します。
- U** 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。

パラメータ	説明
OutputScript	出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。 <b>Input</b> 書き直しを実行せず、入力と同じスクリプトで出力します(デフォルト)。 <b>Native</b> 使用可能な場合は、選択した国のネイティブ スクリプトで出力します。 <b>Latn</b> 英語の値を使用します。
MaximumResults	GetCandidateAddressesLoqate が返す候補住所の最大数。デフォルトは 10 です。最大数は 99 です。

## 応答

GetCandidateAddressesLoqate は、次の出力を返します。

**表 110 : GetCandidateAddressesLoqate の出力**

応答要素	説明
AddressLine1	最初の住所行。
AddressLine2	2 行目の住所行。
AddressLine3	3 行目の住所行。
AddressLine4	4 行目の住所行。
City	都市名。

応答要素	説明
Country	3 文字の ISO 3166-1 Alpha 3 国コード。ISO コードの一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。
FirmName	企業名。
PostalCode	郵便番号。米国では、ZIP Code™になります。
PostalCode.AddOn	ZIP + 4® Code の末尾 4 桁。米国住所にのみ適用されます。
ProcessedBy	住所を処理した住所コーダーを示します。 <b>LOQATE</b> Loqate コーダーが住所を処理しました。
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。次のいずれかの値になります。 • RequestFailed
Status.Description	問題の説明 (ある場合)。次のいずれかの値になります。 <b>Did not return multiples</b> 入力住所はデータベース内の 1 つの住所とのみマッチしました。GetCandidateAddressesLoqate は、一致する可能性のある住所が複数見つかった場合のみデータを返します。

## GetCityStateProvince

GetCityStateProvince は、与えられた入力郵便番号に対する都市および州/省を返します。

注：GetCityStateProvince は、米国およびカナダの住所のみをサポートします。

GetCityStateProvince は、Universal Addressing モジュールに含まれています。

### リソース URL

```
http://server:port/soap/GetCityStateProvince
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetCityStateProvince"
xmlns:spec="http://spectrum.pb.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetCityStateProvinceRequest>
      <get:input_port>
        <get:Input>
          <get:PostalCode>60510</get:PostalCode>
        </get:Input>
      </get:input_port>
    </get:GetCityStateProvinceRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:GetCityStateProvinceResponse
xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/GetCityStateProvince">
      <ns3:output_port>
        <ns3:Result>
          <ns3:ProcessedBy>USA</ns3:ProcessedBy>
          <ns3:PostalCode>60510</ns3:PostalCode>
          <ns3:City>BATAVIA</ns3:City>
          <ns3:City.Type>P</ns3:City.Type>
          <ns3:StateProvince>IL</ns3:StateProvince>
          <ns3:Country>USA</ns3:Country>
          <ns3:user_fields/>
        </ns3:Result>
      </ns3:output_port>
    </ns3:GetCityStateProvinceResponse>
  </soap:Body>
</soap:Envelope>
```

**要求**

## 入力データのパラメータ

入力フィールドを以下の表に示します。

**表 111 : GetCityStateProvince の入力**

パラメータ	説明
PostalCode	ZIP Code™ またはカナダの郵便番号を次のいずれかのフォーマットで示します。 99999 99999-9999 A9A9A9 A9A 9A9

## オプションのパラメータ

**表 112 : GetCityStateProvince のオプション**

パラメータ名	説明
PerformUSProcessing	<p>米国住所を処理するかどうかを指定します。米国住所処理を有効にした場合、GetCityStateProvince は米国住所の州を返します。米国住所処理を無効にすると、米国住所は失敗します。つまり、Status 出力フィールドに "F" が設定されて返されます。出力フィールド Status.Code は、"DisabledCoder" となります。米国住所処理のライセンスを取得していない場合は、ジョブに米国住所が含まれるか否かにかかわらず、米国住所処理を無効にしなければ、ジョブを正常に実行することはできません。</p> <p>注：米国住所を正常に処理するには、米国住所処理の有効なライセンスを取得する必要があります。米国住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、米国住所処理を有効にすると、エラーとなります。</p> <p><b>Y</b>      はい、米国住所を処理します (デフォルト)。</p> <p><b>N</b>      いいえ、米国住所を処理しません。</p>

パラメータ名	説明
Database.US	米国住所処理に使用するデータベースを指定します。Management Console の [米国データベース リソース] パネルで定義されたデータベースのみが使用可能です。
PerformCanadianProcessing	<p>カナダ住所を処理するかどうかを指定します。カナダ住所処理を有効にした場合、GetCityStateProvince はカナダ住所の州を返します。カナダ住所処理を無効にした場合、Status フィールドに "F" が設定されカナダ住所は失敗します。出力フィールド Status.Code は、"DisabledCoder" となります。カナダ住所処理のライセンスを取得していない場合は、ジョブにカナダ住所が含まれるか否かにかかわらず、カナダ住所処理を無効にしなければ、ジョブを正常に実行することはできません。</p> <p>注：カナダ住所を正常に処理するには、カナダ住所処理の有効なライセンスを取得する必要があります。カナダ住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、カナダ住所処理を有効にすると、エラーとなります。</p> <p><b>Y</b>          カナダ住所を処理します (デフォルト)。</p> <p><b>N</b>          カナダ住所を処理しません。</p>
Database.Canada	カナダ住所処理に使用するデータベースを指定します。Management Console の [カナダ データベース リソース] パネルで定義されたデータベースのみが使用可能です。
OutputVanityCity	<p>非正式な都市名を出力に含めるかどうかを指定します。非正式な都市名は、主要都市名の代替名です。例えば、Hollywood は Los Angeles の非正式な都市名です。</p> <p><b>Y</b>          非正式な都市名を含めます。</p> <p><b>N</b>          非正式な都市名を含めません (デフォルト)。</p>
MaximumResults	返される都市と州/省のペアの最大数を指定します。デフォルト値は 10 です。

## 応答

GetCityStateProvince は、入力郵便番号に一致する都市と州/省、およびマッチングの成功または失敗を示すコードを返します。複数の都市/州または都市/省が入力郵便番号にマッチする場合、複数の出力レコードが返されます。

表 113 : GetCityStateProvince の出力

応答要素	説明
City	一致した都市名。
City.Type	USPS® によって正規化された都市名のタイプ (米国住所のみ)。 <ul style="list-style-type: none"> <li><b>V</b> 非正式 (Non-Mailing) 都市名。</li> <li><b>P</b> 主要都市名。都市名は、主要な郵送都市名です。</li> <li><b>S</b> 補助的な都市名。都市名は、代替都市名ですが、許容されます。都市は複数の補助的な都市名を持つことができます。</li> </ul>
PostalCode	入力郵便番号。
ProcessedBy	住所を処理した住所コーダーを示します。次のいずれかです。 <ul style="list-style-type: none"> <li><b>USA</b> 米国住所コーダーが住所を処理しました。</li> <li><b>CAN</b> カナダ住所コーダーが住所を処理しました。</li> </ul>
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <ul style="list-style-type: none"> <li><b>NULL</b> 成功</li> <li><b>F</b> 失敗</li> </ul>
Status.Code	失敗の原因 (ある場合)。次の値のみが有効です。 <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• UnrecognizedPostalCode</li> </ul>



## 応答要素

## 説明

Status.Description

失敗の説明。有効な値は次のとおりです。

**Postal code not found**

Status.Code=UnrecognizedPostalCode の場合にこの値が表示されます。

**PerformUSProcessing disabled**

Status.Code=DisabledCoder の場合にこの値が表示されます。

**PerformCanadianProcessing disabled**

Status.Code=DisabledCoder の場合にこの値が表示されます。

**GetCityStateProvinceLoqate**

GetCityStateProvinceLoqate は、与えられた入力郵便番号に対する都市および州/省を返します。このステージは、Universal Addressing モジュールに含まれています。

## リソース URL

```
http://server:port/soap/GetCityStateProvinceLoqate
```

## 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetCityStateProvinceLoqate"
xmlns:spec="http://spectrum.pb.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetCityStateProvinceLoqateRequest>
      <get:input_port>
        <get:Input>
          <get:PostalCode>60510</get:PostalCode>
          <get:Country>USA</get:Country>
        </get:Input>
      </get:input_port>
    </get:GetCityStateProvinceLoqateRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:GetCityStateProvinceLoqateResponse
xmlns:ns2="http://spectrum.pb.com/"

xmlns:ns3="http://www.pb.com/spectrum/services/GetCityStateProvinceLoqate">

      <ns3:output_port>
        <ns3:Result>
          <ns3:ProcessedBy>LOQATE</ns3:ProcessedBy>
          <ns3:PostalCode>60510</ns3:PostalCode>
          <ns3:City>Batavia</ns3:City>
          <ns3:StateProvince>IL</ns3:StateProvince>
          <ns3:Country>United States</ns3:Country>
          <ns3:Status/>
          <ns3:Status.Code/>
          <ns3:Status.Description/>
          <ns3:user_fields/>
        </ns3:Result>
      </ns3:output_port>
    </ns3:GetCityStateProvinceLoqateResponse>
  </soap:Body>
</soap:Envelope>
```

### 要求

入力データのパラメータ

入力フィールドを以下の表に示します。

表 114 : GetCityStateProvinceLoqate の入力

パラメータ	説明
Country	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2桁の ISO 国コード</li> <li>• 3桁の UPU 国コード</li> <li>• 英語の国名</li> </ul> <p>ISOコードの一覧は、<a href="#">ISO 国コードとモジュールサポート</a> (735ページ) を参照してください。</p>
PostalCode	住所の郵便番号。

## オプション

表 115 : GetCityStateProvinceLoqate のオプション

	説明 / 有効な値						
	住所処理に使用するデータベースを指定します。Management Console の [データベース リソース] パネルで定義されたデータベースのみが使用可能です。						
	GetCityStateProvinceLoqate が返す必要がある住所の最大数。デフォルトは 10 です。						
OutputScript	出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。 <table border="0"> <tr> <td><b>Input</b></td> <td>書き直しを実行せず、入力と同じスクリプトで出力します (デフォルト)。</td> </tr> <tr> <td><b>Native</b></td> <td>使用可能な場合は、選択した国のネイティブ スクリプトで出力します。</td> </tr> <tr> <td><b>Latn</b></td> <td>英語の値を使用します。</td> </tr> </table>	<b>Input</b>	書き直しを実行せず、入力と同じスクリプトで出力します (デフォルト)。	<b>Native</b>	使用可能な場合は、選択した国のネイティブ スクリプトで出力します。	<b>Latn</b>	英語の値を使用します。
<b>Input</b>	書き直しを実行せず、入力と同じスクリプトで出力します (デフォルト)。						
<b>Native</b>	使用可能な場合は、選択した国のネイティブ スクリプトで出力します。						
<b>Latn</b>	英語の値を使用します。						
	データ ライセンス エラーの発生時に Spectrum Technology Platform がどのように応答するかを指定します。 <table border="0"> <tr> <td>ジョブのエラー</td> <td>データ ライセンス エラーが発生した場合、ジョブ全体をエラーにします。</td> </tr> <tr> <td>レコードのエラー</td> <td>データ ライセンス エラーの発生原因となったレコードをエラーにし、処理を続行します。</td> </tr> </table>	ジョブのエラー	データ ライセンス エラーが発生した場合、ジョブ全体をエラーにします。	レコードのエラー	データ ライセンス エラーの発生原因となったレコードをエラーにし、処理を続行します。		
ジョブのエラー	データ ライセンス エラーが発生した場合、ジョブ全体をエラーにします。						
レコードのエラー	データ ライセンス エラーの発生原因となったレコードをエラーにし、処理を続行します。						

## 応答

GetCityStateProvinceLoqate は、入力郵便番号に一致する都市と州/省、およびマッチングの成功または失敗を示すコードを返します。複数の都市/州または都市/省が入力郵便番号にマッチする場合、複数の出力レコードが返されます。

表 116 : GetCityStateProvinceLoqate の出力

応答要素	説明
City	一致した都市名。
Country	で選択した、以下のいずれかのフォーマットで示された国。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>
PostalCode	入力郵便番号。
ProcessedBy	住所を処理した住所コーダーを示します。 <b>LOQATE</b> Loqate コーダーが住所を処理しました。
StateProvince	州または省の省略形。
Status	マッチの成功または失敗。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。次の値のみが有効です。 <ul style="list-style-type: none"> <li>• UnrecognizedPostalCode</li> </ul>
Status.Description	失敗の説明。次の値のみが有効です。 <b>Postal code not found</b> Status.Code=UnrecognizedPostalCode の場合にこの値が表示されます。

## GetPostalCodes

**GetPostalCodes** では、特定の都市の郵便番号の検索が可能です。このサービスは、都市、州、および国を入力として受け取り、その都市の郵便番号を返します。入力を正しい順序で指定しなければ、郵便番号は返されません。

注： **GetPostalCodes** は、米国住所にのみ対応します。

**GetPostalCodes** は、 **Universal Addressing** モジュールに含まれています。

### リソース URL

```
http://server:port/soap/GetPostalCodes
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.pb.com/spectrum/services/GetPostalCodes"
xmlns:spec="http://spectrum.pb.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <get:GetPostalCodesRequest>
      <get:input_port>
        <get:Input>
          <get:City>Holland</get:City>
          <get:StateProvince>MI</get:StateProvince>
        </get:Input>
      </get:input_port>
    </get:GetPostalCodesRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:GetPostalCodesResponse xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/GetPostalCodes">
      <ns3:output_port>
        <ns3:Result>
          <ns3:ProcessedBy>USA</ns3:ProcessedBy>
          <ns3:PostalCode>49422</ns3:PostalCode>
          <ns3:City.Type></ns3:City.Type>
          <ns3:Status/>
          <ns3:Status.Code/>
          <ns3:Status.Description/>
        </ns3:Result>
      </ns3:output_port>
    </ns3:GetPostalCodesResponse>
  </soap:Body>
</soap:Envelope>
```

```

        <ns3:user_fields/>
    </ns3:Result>
    <ns3:Result>
        <ns3:ProcessedBy>USA</ns3:ProcessedBy>
        <ns3:PostalCode>49423</ns3:PostalCode>
        <ns3:City.Type></ns3:City.Type>
        <ns3:Status/>
        <ns3:Status.Code/>
        <ns3:Status.Description/>
        <ns3:user_fields/>
    </ns3:Result>
    <ns3:Result>
        <ns3:ProcessedBy>USA</ns3:ProcessedBy>
        <ns3:PostalCode>49424</ns3:PostalCode>
        <ns3:City.Type></ns3:City.Type>
        <ns3:Status/>
        <ns3:Status.Code/>
        <ns3:Status.Description/>
        <ns3:user_fields/>
    </ns3:Result>
</ns3:output_port>
</ns3:GetPostalCodesResponse>
</soap:Body>
</soap:Envelope>

```

**要求**

## 入力データのパラメータ

GetPostalCodes は、都市、州/省、および国を入力として受け取ります。

**表 117 : GetPostalCodes の入力**

パラメータ	説明
City	<p>郵便番号を検索する都市。</p> <p>City フィールドに都市と州を入力できます。これを行う場合は、StateProvince フィールドを空白のままにする必要があります。</p> <p>City および StateProvince フィールドの文字数が 100 文字を超えてはなりません。</p>
StateProvince	<p>郵便番号を検索する都市の州または省。</p> <p>州は、StateProvince フィールドではなく、City フィールドに入力することもできます。</p> <p>City および StateProvince フィールドの文字数が 100 文字を超えてはなりません。</p>

パラメータ	説明
Country	郵便番号を検索する都市の国コードまたは名前。有効な値は US のみです。

### オプションのパラメータ

表 118 : **GetPostalCodes** のオプション

パラメータ	説明
Database.US	郵便番号検索に使用するデータベースを指定します。Management Console の [米国データベース リソース] パネルで定義されたデータベースのみが使用可能です。
IncludeVanityCity	都市の非正式な都市名の郵便番号を含めるかどうかを指定します。非正式な都市名は、主要都市名の代替名です。例えば、Hollywood は Los Angeles の非正式な都市名です。 <b>Y</b> 非正式な都市名の郵便番号を含めます。 <b>N</b> 非正式な都市名の郵便番号を含めません (デフォルト)。
OutputCityType	都市タイプを出力で返すかどうかを指定します。有効にすると、都市タイプが City.Type フィールドに返されます。 <b>Y</b> 都市タイプを出力に含めます。 <b>N</b> 都市タイプを出力に含めません (デフォルト)。

### 応答

**GetPostalCodes** は、指定された都市の郵便番号を返します。各郵便番号は、以下の表に列挙されたデータとともにそれぞれ個別のレコードで返されます。

表 119 : GetPostalCodes の出力

応答要素	説明
City.Type	<p>USPS® の都市タイプ (米国住所のみ)。都市タイプを判別するには、ZIP Code と都市名を調べます。例えば、メリーランド州ランハムの郵便番号は、20703、20706、および 20784 です。ランハムは、20703 と 20706 では主要都市ですが、20784 では非正式都市です。</p> <p>このフィールド列に値が設定されるのは、OutputCityType=Y の場合のみです。有効な値を次に示します。</p> <p><b>V</b> 非正式 (Non-Mailing) 都市名。</p> <p><b>P</b> 主要都市名。都市名は、主要な郵送都市名です。</p> <p><b>S</b> 補助的な都市名。都市名は、代替都市名ですが、許容されます。都市は複数の補助的な都市名を持つことができます。</p>
PostalCode	指定された都市の郵便番号。
ProcessedBy	このサービスは米国住所に対してのみ機能するため、ProcessedBy には常に USA という 1 つの値が含まれます。
Status	<p>マッチの成功または失敗。</p> <p><b>NULL</b> 成功</p> <p><b>F</b> 失敗</p>
Status.Code	<p>失敗の原因 (ある場合)。次のいずれかです。</p> <ul style="list-style-type: none"> <li>• CountryNotSupported</li> <li>• UnableToLookup</li> </ul>
Status.Description	<p>失敗の説明。</p> <ul style="list-style-type: none"> <li>• 入力された国がサポートされていません</li> <li>• 入力された都市が空白でした</li> <li>• 入力された都市と州 / 省が空白であったか、一致が見つかりませんでした</li> <li>• 都市と州の不一致 (スペルの相違が見つかるか、都市/州が非正式都市であるが、非正式マッチングが許可されていないか、都市/州が ZIP Code と一致しない)</li> </ul>



`ValidateAddress` は、郵便当局の住所データを使用して、住所を正規化し、妥当性を確認します。`ValidateAddress` は、情報を修正し、管轄の郵便当局が推奨する書式で住所の書式を整えることができます。また、郵便番号、都市名、州/省名など、欠落している郵便情報を追加します。

`ValidateAddress` は、`ValidateAddress` が住所の妥当性を確認したかどうか、返した住所の確信レベル、住所の妥当性が確認できなかった場合はその理由など、バリデーション処理に関する結果インジケータも返します。

`ValidateAddress` は、住所のマッチングと正規化において、住所行をコンポーネントに分割し、それらを `Universal Addressing` モジュールの各種データベースの内容と比較します。マッチを検出した場合、入力住所をデータベース情報に合わせて正規化します。データベースにマッチしなかった場合、`ValidateAddress` は、オプションで入力住所の書式を整えます。書式設定プロセスでは、該当する郵便当局の規則に従って住所行の構成を試みます。

`ValidateAddress` は、`Universal Addressing` モジュールに含まれています。

### リソース URL

```
http://server:port/soap/ValidateAddress
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:val="http://www.pb.com/spectrum/services/ValidateAddress">
  <soapenv:Header/>
  <soapenv:Body>
    <val:ValidateAddressRequest>
      <val:input_port>
        <val:Address>
          <val:AddressLine1>1 N. State St.</val:AddressLine1>
          <val:City>Chicago</val:City>
          <val:StateProvince>IL</val:StateProvince>
        </val:Address>
      </val:input_port>
    </val:ValidateAddressRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:ValidateAddressResponse xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/ValidateAddress">
```

```

<ns3:output_port>
  <ns3:Address>
    <ns3:Confidence>87</ns3:Confidence>
    <ns3:RecordType>HighRise</ns3:RecordType>
    <ns3:RecordType.Default>Y</ns3:RecordType.Default>
    <ns3:CountryLevel>A</ns3:CountryLevel>
    <ns3:ProcessedBy>USA</ns3:ProcessedBy>
    <ns3:MatchScore>0</ns3:MatchScore>
    <ns3:AddressLine1>1 N State St</ns3:AddressLine1>
    <ns3:City>Chicago</ns3:City>
    <ns3:StateProvince>IL</ns3:StateProvince>
    <ns3:PostalCode>60602-3302</ns3:PostalCode>
    <ns3:PostalCode.Base>60602</ns3:PostalCode.Base>
    <ns3:PostalCode.AddOn>3302</ns3:PostalCode.AddOn>
    <ns3:Country>United States Of America</ns3:Country>
    <ns3:AdditionalInputData/>
    <ns3:user_fields/>
  </ns3:Address>
</ns3:output_port>
</ns3:ValidateAddressResponse>
</soap:Body>
</soap:Envelope>

```

### 要求

#### 入力データのパラメータ

`ValidateAddress` は、入力として住所を受け取ります。すべての住所は、その住所の国に関係無くこのフォーマットを使用します。米国住所に対する住所行データの処理方法に関する重要な情報については、「[米国住所の住所行処理 \(572ページ\)](#)」を参照してください。

表 120 : 入力フォーマット

パラメータ	書式	説明
AddressLine1	文字列 [50]	最初の住所行。
AddressLine2	文字列 [50]	2 行目の住所行。
AddressLine3	文字列 [50]	3 行目の住所行。 カナダの住所には適用されません。

パラメータ	書式	説明
AddressLine4	文字列 [50]	4 行目の住所行。 カナダの住所には適用されません。
AddressLine5	文字列 [50]	5 行目の住所行。 英国住所にのみ適用されます。通り名、ユニット番号、ビルディング番号などを含めることができます。
City	文字列 [50]	都市名。 米国住所に限り、都市、州、および ZIP Code™ を City フィールドに入力することができます。これを行う場合は、StateProvince フィールドと PostalCode フィールドを空白のままにする必要があります。
StateProvince	文字列 [50]	州または省。 米国の住所に限り、州を StateProvince フィールドの代わりに City フィールドに入れることができます。
PostalCode	文字列 [10]	住所に対する郵便番号を次のいずれかのフォーマットで示します。 99999 99999-9999 A9A9A9 A9A 9A9 9999 999  米国式の住所の場合に、住所に限り、ZIP Code™ を City フィールドに入力することができます。  米国式の住所の場合に、住所に限り、都市、州、または ZIP Code™ が PostalCode フィールドにある場合に、ValidateAddress がデータをパースングして、住所を正しく処理することができます。最適な結果を得るため、このデータを適切なフィールド (City、StateProvince、PostalCode) に入力してください。

パラメータ	書式	説明
Country	文字列 [50]	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2 文字の ISO 3166-1 Alpha 2 国コード</li> <li>• 3 文字の ISO 3166-1 Alpha 3 国コード</li> <li>• 英語の国名</li> <li>• フランス語の国名</li> <li>• ドイツ語の国名</li> <li>• スペイン語の国名</li> </ul> <p>ISO コードの一覧は、<a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。</p>
FirmName	文字列 [50]	会社名または企業名。
USUrbanName	文字列 [50]	米国住所都市化名。主にプエルトリコの住所で使用されます。
CustomerID	文字列 [9]	この郵便物が汎用バーコードを使用している場合、USPS®が割り当てた顧客 ID をこのフィールドに指定します。ValidateAddress の汎用バーコードは、OneCode ACS®サービスを使用する郵便物に使用されます。
CanLanguage	文字列	<p>カナダの住所に限り、CanFrenchFormat=T オプションが使用されている場合に、住所が英語かフランス語かを示します。</p> <p>このフィールドが空白の場合、アドレスは英語でフォーマットされています。このフィールドに空白以外の値が含まれる場合、住所はフランス語でフォーマットされています。ケベックの住所は、このフィールドの値に関係なく常にフランス語でフォーマットされます。</p>

## 米国住所の住所行処理

米国住所の場合、AddressLine1 から AddressLine4 の入力フィールドの処理方法は、企業名抽出または都市化コード抽出のオプションが有効になっているかどうかによって異なります。2つのオプションのいずれかが有効になっている場合は、ValidateAddress は、4 つすべてのフィールドのデータを参照して、住所の妥当性を確認し、要求されたデータ (企業名または都市化コード) を抽出します。どちらのオプションも有効でない場合は、ValidateAddress は、空白でない最初の 2

つの住所行フィールドのみを使用して、妥当性を確認します。他の住所行フィールドのデータは、AdditionalInputData 出力フィールドに返されます。例を次に示します。

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

この住所において、企業名抽出または都市化コード抽出が有効である場合は、ValidateAddress は、4 つすべての住所行を確認します。企業名抽出と都市化コード抽出がどちらも有効でない場合は、ValidateAddress は、AddressLine1 と AddressLine3 (空白でない最初の 2 つの住所行) を参照して、そのデータを使用して住所の妥当性を確認します。AddressLine4 のデータは、AdditionalInputData 出力フィールドに返されます。

オプションのパラメータ

出力データ オプション

以下の表に、ValidateAddress が返す情報の種類を制御するオプションの一覧を示します。これらのオプションのうちの一部は、カナダ住所に対してオーバーライドすることができます。詳細については、[カナダ住所のオプション](#) (604ページ) を参照してください。

表 121 : 出力データ オプション

パラメータ	説明
OutputRecordType	<p>出力レコードのタイプ。1つ以上の場合はリストで提供されます。</p> <ul style="list-style-type: none"> <li data-bbox="690 493 1421 871"><b>A</b> 住所データの1~4行に加えて、都市、州、郵便番号、企業名、および都市化名情報を返します。各住所行は、封筒に記載される住所の実際の行に対応しています。詳細については、<a href="#">応答 (616ページ)</a> を参照してください。住所の妥当性を確認できた場合は、住所行には正規化済み住所が含まれます。正規化済み住所では、句読文字が取り除かれ、方向指示と通り接尾語には省略形が使用され、住所要素が修正されています。住所の妥当性を確認できなかった場合は、住所行には入力住所がそのまま含まれます ("パス スルー" データ)。OutputRecordType=A を指定しなかった場合でも、妥当性が確認されなかった住所は必ず、パス スルー データとして住所行フィールドに含まれます。</li> <li data-bbox="690 903 1421 1102"><b>E</b> パース済み住所要素。家番号、通り名、通り接尾語、方向指示などの住所の各要素が、個別のフィールドに返されます。詳細については、<a href="#">#unique_113</a>を参照してください。"E" を指定し、OutputFormattedOnFail=Y を指定した場合は、パース済み住所要素に、妥当性が確認できなかった住所の入力住所が含まれることとなります。</li> <li data-bbox="690 1134 1421 1480"><b>I</b> パース済み入力。このオプションでは、住所の妥当性を確認できたかどうかにかかわらず、入力住所をパース済み形式で返します。家番号、通り名、通り接尾語、方向指示などの入力住所の各要素が、個別のフィールドに返されます。パース済み入力 (値 "I") は、妥当性を確認できなかった入力だけでなく、すべての入力住所をパース済み形式で返す点で、OutputRecordType=E と OutputFormattedOnFail=Y の組み合わせとは異なります。詳細については、<a href="#">パース済み入力 (620ページ)</a> を参照してください。</li> <li data-bbox="690 1501 1421 1596"><b>P</b> 郵便データ。出力住所には、妥当性が確認された各住所の追加データが含まれます。詳細については、<a href="#">郵便データ出力 (623ページ)</a> を参照してください。</li> </ul> <p><b>空</b> 住所データまたは郵便データを返しません。 <b>白</b></p>

## パラメータ

## 説明

---

**OutputFieldLevelReturnCodes**

フィールドレベルの結果インジケータを含めるかどうかを指定します。フィールドレベルの結果インジケータは、各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、HouseNumber のフィールドレベルの結果インジケータは **HouseNumber.Result** に格納されます。結果インジケータの出力フィールドの完全な一覧は、[フィールドレベルの結果インジケータ \(629ページ\)](#) を参照してください。

- N**     フィールドレベルのリターン コードを出力しません (デフォルト)。
  - Y**     フィールドレベルのリターン コードを出力します。
-

パラメータ

説明

---

OutputFormattedOnFail



## パラメータ

## 説明

住所の妥当性を確認できない場合に書式を整えた住所を返すかどうかを指定します。住所には、その国の標準住所書式が設定されます。このオプションを選択しない場合、住所の妥当性を確認できないと、出力住所フィールドは空白になります。

注：このオプションは、米国およびカナダの住所のみに適用されます。その他の住所に対して書式を整えたデータは返されません。

**N** 失敗した住所の書式を整えません (デフォルト)。

**Y** 失敗した住所の書式を整えます。

OutputRecordType オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。OutputRecordType=E を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。ValidateAddress が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、OutputRecordType=I を指定します。

Option.OutputRecordType オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。Option.OutputRecordType=E を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。ValidateAddress が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、Option.OutputRecordType=I を指定します。

**[標準住所を含める]**、**[住所行の要素を含める]**、および **[郵便情報を含める]** の各チェック ボックスで指定されたフォーマットを使用して、書式を整えた住所が返されます。**[住所行の要素を含める]** を選択した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。ValidateAddress が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**[正規化された入力住所要素を含める]** を選択します。

Y を指定する場合は、OutputRecordType に対して "A" と "E" またはそのいずれかを指定する必要があります。

## パラメータ

## 説明

Y を指定する場合は、Option.OutputRecordType に対して "A" と "E" またはそのいずれかを指定する必要があります。

このオプションをオンにする場合は、**[標準住所を含める]** と **[住所行の要素を含める]** またはそのいずれかを選択する必要があります。

## OutputStreetNameAlias

米国住所に対しては、通り名のエイリアスを出力に使用するかどうかを指定します。通り名のエイリアスとは、通りに対する別名で、通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。基本名とは、通り全体に適用される名称です。

- N** 出力において、通り名のエイリアスを返しません。
- Y** 通り名のエイリアスが存在する場合は、出力において、通り名のエイリアスを返します (デフォルト)。

## OutputStreetNameAlias

米国住所に対しては、入力に使用されている通り名のエイリアスの処理方法を指定します。通り名のエイリアスとは、通りに対する別名で、通常は、通り上の特定の範囲の住所のみに対して使用されます。

このオプションを有効にすると、入力に使用されている通り名のエイリアスが、出力にも使用されます。このオプションを有効にしない場合は、入力に使用されている通り名のエイリアスが、出力では基本通り名に変換されますが、次の例外があります。

- 入力において、よく使用されるエイリアスが使用されている場合は、そのエイリアスが必ず出力で使用されます。
- 入力で使用されている変更名のエイリアスは、出力では必ず基本通り名に変換されます。

これは、ValidateAddress で通り名のエイリアスを処理する方法を制御する 3 つのオプションのうちの 1 つです。他の 2 つは OutputPreferredAlias と OutputAbbreviatedAlias です。

注: OutputAbbreviatedAlias を有効にした場合は、OutputStreetNameAlias を無効にした場合でも、必ず省略形エイリアスが出力に使用されます。

- N** 出力において、通り名のエイリアスを返しません。
- Y** 入力された通り名がエイリアスである場合は、出力において、通り名のエイリアスを返します (デフォルト)。

## パラメータ

## 説明

---

**OutputAddressBlocks**

実際の郵便物に印字される、書式を整えた住所を返すかどうかを指定します。住所の各行が、別々の住所ブロックフィールドに入れて返されます。**AddressBlock1** から **AddressBlock9** まで、最大 9 つの住所ブロック出力フィールドが使用されます。

例えば、以下の住所入力の場合、

AddressLine1: 4200 Parliament Place  
AddressLine2: Suite 600  
City: Lanham  
StateProvince: MD  
PostalCode: 20706

以下の住所ブロックが出力されます。

AddressBlock1: 4200 PARLIAMENT PL STE 600  
AddressBlock2: LANHAM MD 20706-1882  
AddressBlock3: UNITED STATES OF AMERICA

**ValidateAddress** は、郵便当局の規格に従って、住所の書式を整えて、住所ブロックの形式にします。国名は、万国郵便連合 (UPU) の国名で返されます。**OutputCountryFormat** オプションは、住所ブロックの国名には影響を与えないことに注意してください。このオプションは、**Country** 出力フィールドに返される名前だけに影響を与えます。

米国およびカナダ以外の住所に対しては、**ValidateAddress** で住所の妥当性が確認できなかった場合、住所ブロックは返されません。米国およびカナダの住所に対しては、妥当性が確認できなかった場合も住所ブロックが返されます。

- N** 住所ブロックを返しません。こちらがデフォルトです。
- Y** 住所ブロックを返します。

## パラメータ

## 説明

## OutputAMAS

実際の郵便物に印字される、書式を整えた住所を返すかどうかを指定します。住所の各行が、別々の住所ブロックフィールドに入れて返されます。**AddressBlock1** から **AddressBlock9** まで、最大 9 つの住所ブロック出力フィールドが使用されます。

例えば、以下の住所入力の場合、

AddressLine1: 4200 Parliament Place  
 AddressLine2: Suite 600  
 City: Lanham  
 StateProvince: MD  
 PostalCode: 20706

以下の住所ブロックが出力されます。

AddressBlock1: 4200 PARLIAMENT PL STE 600  
 AddressBlock2: LANHAM MD 20706-1882  
 AddressBlock3: UNITED STATES OF AMERICA

**ValidateAddress** は、郵便当局の規格に従って、住所の書式を整えて、住所ブロックの形式にします。国名は、万国郵便連合 (UPU) の国名で返されます。OutputCountryFormat オプションは、住所ブロックの国名には影響を与えないことに注意してください。このオプションは、**Country** 出力フィールドに返される名前だけに影響を与えます。

米国およびカナダ以外の住所に対しては、**ValidateAddress** で住所の妥当性が確認できなかった場合、住所ブロックは返されません。米国およびカナダの住所に対しては、妥当性が確認できなかった場合も住所ブロックが返されます。

- N** 住所ブロックを返しません。こちらがデフォルトです。
- Y** 住所ブロックを返します。

## 下院選挙区の取得

**ValidateAddress** は、住所に対する米下院選挙区を特定できます。

下院選挙区を取得するには、OutputRecordType に **P** を含める必要があります。

OutputRecordType の詳細については、[出力データオプション](#) (573ページ) を参照してください。

表 122 : 下院選挙区出力

応答要素	説明
USCongressionalDistrict	下院選挙区番号。住所が州以外の住所 (プエルトリコやワシントン D.C. など) である場合は、このフィールドは空白になります。

#### 郡名の取得

ValidateAddress は、ある住所が所在する郡を特定し、その郡名を返すことができます。

注：郡名は、米国住所にのみ適用されます。

郡名を取得するには、OutputRecordType に P を含める必要があります。OutputRecordType の詳細については、[出力データ オプション \(573ページ\)](#) を参照してください。

表 123 : 郡名出力

応答要素	説明
USCountyName	郡名

#### FIPS 郡番号の取得

連邦情報処理標準 (FIPS) 郡番号は、州の中の各郡を識別する番号です。これらの番号は、州レベルにおいてのみ一意であり、国レベルでは一意ではないことに注意してください。詳細については、<http://www.census.gov> を参照してください。

注：FIPS 郡番号は、米国住所にのみ適用されます。

FIPS 郡番号を取得するには、OutputRecordType に P を含める必要があります。OutputRecordType の詳細については、[出力データ オプション \(573ページ\)](#) を参照してください。

表 124 : FIPS 郡番号出力

	説明
USFIPSCountyNumber	FIPS (連邦情報処理標準) 郡番号

#### 配達ルート コードの取得

配達ルート コードとは、個々の郵便配達者に割り当てられた一意の識別子で、これによって米国の各配達ルートを一意に識別することができます。ValidateAddress は、宛先の配達ルートを表すコードを返すことができます。

注：配達ルート コードは、米国住所に対してのみ存在します。

配達ルート コードを取得するには、OutputRecordType に P を含める必要があります。OutputRecordType の詳細については、[出力データオプション](#) (573ページ) を参照してください。

表 125 : 配達ルート コード出力

応答要素	説明
USCarrierRouteCode	配達ルート コード

#### 配達ポイント バーコードの作成

配達ポイント バーコード (DPBC) は、住所を POSTNET™ バーコードで表記したものです。開始および終了フレームバーと、ZIP+4® Code、ストリートの住所の番号に基づいて計算された値、および修正ディジットの 1 桁ごとに 5 本のバーがあり、合計 62 本のバーで構成されます。DPBC により、配達業者の徒歩経路のレベルにまで手紙を自動仕分けすることができます。ValidateAddress は、DPBC の作成に必要なデータを生成します。

注：配達ポイント バーコードは、米国住所に対してのみ存在します。配達ポイント バーコードの詳細については、<http://www.usps.com> を参照してください。

DPBC の作成に必要なデータを生成するには、OutputRecordType に P を含める必要があります。OutputRecordType の詳細については、[出力データオプション](#) (573ページ) を参照してください。

表 126 : 配達ポイント バーコード出力

	説明
PostalBarCode	配達ポイント バーコードの配達ポイント部分
USBCCheckDigit	11桁の配達ポイント バーコードのチェック デジット部分

DPBC を作成するには、ValidateAddress 出力の値を次のように結合します。

PostalCode.Base + PostalCode.Addon + PostalBarcode + USBCCheckDigit

例えば、次のデータがあるとしてします。

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

このデータから作成されるバーコードは、次のようになります。

494234506292

#### デフォルト オプション

以下の表に、住所のフォーマットと処理を制御するオプションの一覧を示します。これらのオプションはデフォルトですべての住所に適用されるので、"デフォルト オプション" と言います。これらのオプションのうちの一部は、カナダ住所に対してオーバーライドすることができます。詳細については、「[カナダ住所のオプション \(604ページ\)](#)」を参照してください。

表 127 : デフォルト オプション

パラメータ	説明
OutputCasing	出力住所の大文字と小文字の区別を指定します。次のいずれかです。 <b>M</b> 出力に大文字と小文字を混在させます (デフォルト)。例: 123 Main StMytown FL 12345 <b>U</b> 出力に大文字を使用します。例: 123 MAIN STMYTOWN FL 12345

パラメータ	説明
OutputPostalCodeSeparator	<p>ZIP™ Code またはカナダの郵便番号に区切り文字 (スペースまたはハイフン) を使用するかどうかを指定します。</p> <p>例えば、区切り文字ありの ZIP + 4® Code は 20706-1844、区切り文字なしは 207061844 になります。区切り文字ありのカナダの郵便番号は P5E"1S7、区切り文字なしは P5E1S7 になります。</p> <p><b>Y</b> 区切り文字を使用します (デフォルト)。</p> <p><b>N</b> 区切り文字を使用しません。</p> <p>注: カナダの郵便番号ではスペースが、米国の ZIP + 4® コードではハイフンが使用されます。</p>
OutputMultinationalCharacters	<p>ウムラウト記号やアクセント記号などの付加記号を含む多国籍文字を返すかどうかを指定します(米国住所ではサポートされません)。</p> <p><b>N</b> 出力に多国籍文字を使用しません(デフォルト)。標準の ASCII 文字のみが返されます。</p> <p><b>Y</b> 出力に多国籍文字を使用します。</p>
KeepMultimatch	<p>一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを示します。</p> <p><b>Y</b> 複数のマッチを返します (デフォルト)。</p> <p><b>N</b> 複数のマッチを返しません。</p> <p>詳細については、「<a href="#">複数マッチを返す (589ページ)</a>」を参照してください。</p>



## パラメータ

## 説明

---

**StandardAddressFormat**

米国住所の補助的な住所情報を配置する場所を指定します。補助的な住所情報とは、部屋番号やアパート番号などの指定子のことです。例えば、次の住所の補助的な住所情報は "Apt 10E" で、主要な住所情報は "424 Washington Blvd" です。

Apt 10E424 Washington BlvdSpringfield MI 49423

- C** 主要な住所情報と補助的な住所情報の両方を **AddressLine1**に配置します (デフォルト)。
- S** 主要な住所情報を **AddressLine1**に、補助的な住所情報を **AddressLine2** に配置します。
- D** 主要な住所情報と補助的な住所情報の両方を **AddressLine1**に配置し、二重住所からドロップされた情報を **AddressLine2** に配置します。二重住所とは、通り情報と、PO Box/地方配送路/幹線請負契約情報の両方を含む住所のことです。詳細については、「[二重住所ロジックについて \(588ページ\)](#)」を参照してください。

---

**OutputShortCityName**

短い都市名または非正式な都市名を代替名として持つ都市名を書式設定する方法を指定します。米国およびカナダの住所に適用されます。

- Y** **USPS**<sup>®</sup> が承認した都市の略称が 1 つある場合、それを返します。**USPS**<sup>®</sup> は、14 文字以上の都市名に対し、略称を定めています。都市の略称は 13 文字以下で、宛名ラベルのサイズが限られている場合に使用できます。短い都市名が存在しない都市に対しては、正式な都市名が返されます。
  - N** 長い都市名を返します (デフォルト)。
  - S** 入力住所に省略された都市名が使用されている場合にのみ、省略された都市名を返します。入力住所に短い都市名が使用されていない場合は、その都市に対する **USPS**<sup>®</sup> の規則によって、長い都市名または短い都市名が返されます。**CASS**<sup>™</sup> テストを実行する場合は、このオプションを選択します。
  - V** 入力都市名が非正式な都市名の場合、非正式な都市名 (非正式名) を出力します。例えば、"Hollywood" は "Los Angeles" の非正式な都市名です。このオプションを選択せず、入力都市名が非正式な都市名の場合、長いバージョンの郵送都市名が返されます。
-

## パラメータ

## 説明

---

**OutputCountryFormat**

**Country** 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。

- E** 英語の国名を使用します (デフォルト)。
  - S** スペイン語の国名を使用します。
  - F** フランス語の国名を使用します。
  - G** ドイツ語の国名を使用します。
  - I** 国名の代わりにその国の2文字のISOの略語を使用します。
  - U** 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。
-

## パラメータ

## 説明

## HomeCountry

デフォルトの国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がカナダにある場合は、カナダを指定します。ValidateAddress は、[StateProvince]、[PostalCode]、および [Country] の各住所フィールドから国を特定できなかった場合、指定された国を使用して、住所の検証を試みます。有効な国名には次のものがあります。

Afghanistan, Albania, Algeria, American Samoa, 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, Equatorial 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, Suriname, 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

## パラメータ

## 説明

## DualAddressLogic

複数の空白ではない住所行があるか、複数の住所タイプが同じ住所行に設定されている場合にどのタイプの一致を返すかを指定します。(米国住所のみ)。

**N** (デフォルト)USPS®CASS™ の規則では、以下の優先順位に基づいて、返す住所を決定します。

1. PO Box
2. Firm
3. Highrise
4. Street
5. Rural Route
6. General Delivery

**S** 住所行に関係なく、通り一致を返します。

**P** 住所行に関係なく、PO Box 一致を返します。

詳細については、「[二重住所ロジックについて \(588ページ\)](#)」を参照してください。

## 二重住所ロジックについて

米国住所に対してのみ、DualAddressLogic オプションは、通り情報と PO Box/地方配送路/幹線請負契約情報の両方が住所に含まれる場合に、Validate Address が通り一致を返すか、または PO Box/地方配送路/幹線請負契約一致を返すかを制御します。

注：通り情報が PO Box/地方配送路/幹線請負契約情報と別の住所行入力フィールドに含まれている場合、DualAddressLogic オプションは、効果を持ちません。

例えば、次の入力住所が与えられたとします。

AddressLine1: 401 N Main St Apt 1 POB 1City: KempStateProvince: TXPostalCode: 75143

ValidateAddress は、次のいずれかを返します。

- DualAddressLogic が N または P のいずれかに設定されている場合は、次を返します。

AddressLine1: PO Box 1City: KempStateProvince: TXPostalCode: 75143-0001

- DualAddressLogic が S に設定されている場合は、次を返します。

AddressLine1: 401 N Main St Apt 1City: KempStateProvince: TXPostalCode: 75143-4806

住所の正規化に使用されない住所データは、次の 2 カ所のいずれかに返すことができます。

- **AddressLine2 — StandardAddressFormat=D** を指定した場合、住所の正規化に使用されない住所情報は AddressLine2 フィールドに返されます。詳細については、[デフォルト オプション \(583ページ\)](#) を参照してください。例えば、二重住所に対して通り一致を返すと選択すると、次のようになります。

AddressLine1: 401 N Main St Apt 1AddressLine2: PO Box 1City: KempStateProvince:  
TXPostalCode: 75143-0001

- **AdditionalInputData** — StandardAddressFormat=D を指定しない場合、住所の正規化に使用されない住所情報は **AdditionalInputData** フィールドに返されます。このオプションの詳細については、[デフォルト オプション \(583ページ\)](#) を参照してください。例えば、二重住所に対して通り一致を返すと選択すると、次のようになります。

AddressLine1: 401 N Main St Apt 1City: KempStateProvince: TXPostalCode:  
75143-0001AdditionalInputData: PO Box 1

ドロップされた住所情報を取得するには、StandardAddressFormat オプションを **D** に設定します。詳細については、[デフォルト オプション \(583ページ\)](#) を参照してください。

#### 複数マッチを返す

ValidateAddress が、入力住所に一致する可能性のある複数の住所を郵便データベース内で検出した場合、一致する可能性のある住所を ValidateAddress が返すよう設定できます。例えば、次の住所は米国郵便データベース内の複数の住所にマッチします。

PO BOX 1  
New York, NY

#### オプション

複数マッチを返すには、次の表に示すオプションを使用します。

**表 128 : 複数マッチのオプション**

パラメータ	説明
KeepMultimatch	一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを示します。
<b>Y</b>	複数のマッチを返します (デフォルト)。
<b>N</b>	複数のマッチを返しません。

パラメータ	説明
MaximumResults	返す住所の最大数を示す 1 ~ 10 の数字を入力します。 デフォルト値は 1 です。  注：Keepmultimatch=N と KeepMultimatch=Y/MaximumResults=1 の違いは、KeepMultimatch=N は複数マッチによって失敗が返され、KeepMultimatch=Y かつ MaximumResults=1 は複数マッチによって 1 つのレコードが返される点です。
OutputFieldLevelReturnCodes	どの出力住所が候補住所かを特定するには、OutputFieldLevelReturnCodes に対して値 Y を指定する必要があります。このように設定すると、候補住所のレコードのフィールドレベルの結果インジケータに 1 つ以上の値 "M" が格納されます。

## 出力

複数マッチを返すよう選択した場合、住所は指定した住所フォーマットで返されます。住所フォーマットの指定については、[出力データオプション](#) (573ページ) を参照してください。どのレコードが候補住所であるかを特定するには、フィールドレベルの結果インジケータに複数の値 "M" があるかどうかを調べます。詳細については、[フィールドレベルの結果インジケータ](#) (629ページ) を参照してください。

### 米国住所のオプション

## パラメータ

## 説明

---

**PerformUSProcessing**

米国住所を処理するかどうかを指定します。米国住所処理を有効にすると、**ValidateAddress** は米国住所の検証を試みます。米国住所処理を無効にすると、米国住所は失敗します。つまり、**Status** 出力フィールドに "F" が設定されて返されます。出力フィールド **Status.Code** は、"DisabledCoder" となります。米国住所処理のライセンスを取得していない場合は、ジョブに米国住所が含まれるか否かにかかわらず、米国住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：米国住所を正常に処理するには、米国住所処理の有効なライセンスを取得する必要があります。米国住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、米国住所処理を有効にすると、エラーとなります。

- N**      いいえ、米国住所を処理しません。
- Y**      はい、米国住所を処理します。デフォルト

---

**Database.US**

米国住所の妥当性の確認に使用するデータベースを指定します。**Management Console** の [米国データベース リソース] パネルで定義されたデータベースのみが使用可能です。

---

**PerformLOT**

**Enhanced Line of Travel (eLOT)** 処理は、トラベルライン シーケンス コードを住所に割り当てます。住所は eLOT 順でソートされませんが、住所を eLOT 順にソートするために使えるトラベルライン シーケンス コードが提供されます。

eLOT 処理を実行するには、eLOT データベースをインストールしておく必要があります。

- N**      トラベルライン処理を実行しません。デフォルト
- Y**      トラベルライン処理を実行します。

このオプションで返される出力フィールドの一覧については、[Enhanced Line of Travel 出力 \(647ページ\)](#) を参照してください。

---

パラメータ	説明
PerformRDI	<p>Residential Delivery Indicator (RDI™) 処理は、住所が (企業住所ではなく) 個人住所であることを調べます。RDI™ 処理を実行するには、RDI™ データベースをインストールしておく必要があります。</p> <p>DPV® と RDI™ の両方の処理を有効にしている場合は、RDI™ 情報は、住所が有効な配達ポイントである場合のみ返されます。DPV® で住所の妥当性が確認されなかった場合は、RDI™ データは返されません。</p> <p><b>N</b> Residential Delivery Indicator 処理を実行しません。デフォルト</p> <p><b>Y</b> Residential Delivery Indicator 処理を実行します。</p>
PerformESM	<p>Enhanced Street Matching (ESM) は、追加のマッチング ロジックを適用することにより、綴りに誤りがある通り名や複雑な通り名を修正し、マッチ結果を得ます。ESMを使うと、より多くの住所の妥当性を確認できるようになりますが、パフォーマンスは低下します。ASMが有効な場合は、ESMを実行することはできません。</p> <p><b>N</b> Enhanced Street Matching を実行しません。デフォルト</p> <p><b>Y</b> Enhanced Street Matching を実行します。</p>
PerformASM	<p>All Street Matching (ASM) は、ESM 処理に加えて追加のマッチング ロジックを適用することにより、通り名の誤りを修正し、マッチ結果を得ます。通りの最初の文字が誤っている場合の通りのマッチングに有効です。ASMは、最良の住所検証結果を提供しますが、パフォーマンスは低下します。</p> <p><b>N</b> All Street Matching を実行しません。</p> <p><b>Y</b> All Street Matching を実行します。デフォルト</p>



## パラメータ

## 説明

## PerformDPV

Delivery Point Validation (DPV<sup>®</sup>) は、特定の住所が有効な住所の範囲内にあるかどうかを確認するのではなく、特定の住所が存在するかどうかを確認します。CMRA 処理は、住所が、Commercial Mail Receiving Agency (CMRA: 民間私書箱) と呼ばれる民間企業が貸し出す私書箱であるかどうかを確認します。

DPV および CMRA 処理を実行するには、DPV データベースをインストールしておく必要があります。DPV データベースには、DPV と CMRA の両方のデータが含まれます。

**N** Delivery Point Validation または CMRA 処理を実行しません。デフォルト

**Y** Delivery Point Validation または CMRA 処理を実行します。

このオプションで返される出力フィールドの一覧については、[DPV および CMRA 出力](#) (650ページ) を参照してください。

## PerformLACSLink

USPS<sup>®</sup> Locatable Address Conversion System (LACS) は、地方配送路の住所を通り名に沿った住所に変換した場合、PO Box 番号の再割り当てがあった場合、または通り名に沿った住所が変更した場合に、それに伴って変更した住所を修正します。LACS<sup>Link</sup> 処理を有効にした場合、妥当性が確認できなかった住所、または妥当性が確認され、LACS<sup>Link</sup> 変換のフラグが付けられた住所に対し、その処理が実行されます。

LACS<sup>Link</sup> 処理を実行するには、LACS<sup>Link</sup> データベースをインストールしておく必要があります。

**N** LACS<sup>Link</sup> 変換を行いません。デフォルト

**Y** LACS<sup>Link</sup> 変換を行います。

このオプションで返される出力フィールドの一覧については、[LACSLink 出力](#) (648ページ) を参照してください。

## パラメータ

## 説明

---

**PerformEWS**

Early Warning System (EWS) は、USPS® EWS ファイルを使用して、ZIP + 4® データベースには存在しない住所の妥当性を確認します。

EWS 処理を実行するには、EWS データベースをインストールしておく必要があります。

入力住所が、EWS ファイルの住所に一致する場合、次のレコード レベルの結果インジケータが返されます。

- Status="F"
- Status.Code="EWSFailure"
- Status.Description="Address found in EWS table"

**N** EWS 処理を実行しません。デフォルト

**Y** EWS 処理を実行します。

---

パラメータ

説明

---

ExtractFirm

## パラメータ

## 説明

AddressLine1 ~ AddressLine4 から企業名を抽出し、FirmName 出力フィールドに入れるかどうかを指定します。このオプションは、入力レコードの FirmName フィールドが空白で、住所行が複数存在する場合に適用されます。

- Y**            企業名を抽出します。
- N**            企業名を抽出しません。デフォルト

住所行の中の企業名を特定するため、住所行をスキャンし、どのフィールドが住所行で、どのフィールドが企業名の行であるかを特定するためのキーワードおよびパターンが検索されます。この処理はパターンに基づいて行われるため、フィールドが誤って認識される場合があります。最適な企業名抽出を行うためのヒントを、以下に示します。

- 可能ならば、主要な住所要素を AddressLine1、補助的な要素を AddressLine2、都市化名を AddressLine3、企業名を AddressLine4 に配置します。住所に都市化コードが存在しない場合は、企業名を AddressLine3 に配置し、AddressLine4 を空白にします。例を次に示します。

**AddressLine1:** 4200 Parliament Place**AddressLine2:** Suite 600**AddressLine3:** Pitney Bowes  
**AddressLine4:** <blank>

- 住所行を 2 行だけ定義する場合は、ほとんどの場合 AddressLine2 には補助的な住所が入ります。AddressLine2 を企業名として処理する確率を上げるには、企業名を AddressLine3 に配置し、AddressLine2 は空白にします。
- 企業名に数字が含まれていると ("1 Stop Software" の "1" など)、そのフィールドが住所行として扱われる可能性が高くなります。

以下に、企業名抽出の例をいくつか示します。

- 次の例では、AddressLine2 が FirmName 出力フィールドに抽出されます。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place Suite 600**AddressLine2:** International Goose Feathers inc.

- 次の例では、AddressLine3 が FirmName 出力フィールドに抽出されます。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place**AddressLine2:** Suite 600**AddressLine3:** Pitney Bowes

- 次の例では、AddressLine3 は AdditionalInputData 出力フィールドに配置されます。FirmName 入力フィールドが空白ではないため、企業名は抽出されません。

**FirmName:** International Goose Feathers Inc.**AddressLine1:** 4200 Parliament Place**AddressLine2:** Suite 600**AddressLine3:** Pitney Bowes

- 次の例では、空白でない住所行が 1 行しかなく、その行は必ず主要な住所要素として処理されるため、企業名は抽出されません。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place Suite 600

## パラメータ

## 説明

- 次の例では、AddressLine2は補助的な住所要素として処理されます。数字の"1"が含まれているために、そのフィールドが補助的な住所要素として扱われるためです。

**FirmName:** <空白>**AddressLine1:** 4200 Parliament Place Suite  
600**AddressLine2:** 1 Stop Software

## ExtractUrb

AddressLine1～AddressLine4から都市化名を抽出し、USUrbanName出力フィールドに入れるかどうかを指定します。このオプションは、入力レコードのUSUrbanNameフィールドが空白で、住所行が複数存在する場合に適用されます。

**Y** 都市化名を抽出します。

**N** 都市化名を抽出しません。デフォルト

住所行の中の都市化名を特定するため、住所行をスキャンし、どのフィールドが住所行で、どのフィールドが都市化名の行であるかを特定するためのキーワードおよびパターンが検索されます。この処理はパターンに基づいて行われるため、フィールドを誤って認識する場合があります。最適な都市化名抽出を行うには、できる限り、主要な住所要素をAddressLine1、補助的な要素をAddressLine2、都市化名をAddressLine3、企業名をAddressLine4に配置します。例を次に示します。

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

## パラメータ

## 説明

## PerformSuiteLink

Suite<sup>Link™</sup> 処理を実行するかどうかを指定します。

Suite<sup>Link</sup> は、補助的な住所情報の妥当性が確認できなかった米国の企業住所に対し、その補助的な住所情報を修正します。Suite<sup>Link</sup> 処理が有効になっている場合、既知の企業名とその補助的な住所情報からなるデータベースに対して、企業名のマッチングが行われます。

例を次に示します。

企業名: Pitney Bowes  
住所行 1: 4200 Parliament Place  
Address Line 2: STE 1  
郵便番号: 20706

この場合、Suite<sup>Link</sup> は、Suite 番号を以下の正しい Suite 番号に変更します。

企業名: Pitney Bowes  
住所行 1: 4200 Parliament Place  
Address Line 2: **STE 600**  
Postal Code: 20706-1844

Suite<sup>Link™</sup> 処理を実行するには、Suite<sup>Link™</sup> データベースをインストールしておく必要があります。

このオプションは、次の値のいずれかを取ります。

- N** Suite<sup>Link™</sup> を使用しません。デフォルト
- Y** Suite<sup>Link™</sup> を使用します。

このオプションで返されるフィールドの一覧については、[SuiteLink 出力 \(652ページ\)](#) を参照してください。

## パラメータ

## 説明

## OutputPreferredAlias

通り名に対してよく使用されるエイリアスを出力に使用するかどうかを指定します。

米国における通り名のエイリアスとは、通りの一部に付けられた別名のことです。通り名のエイリアスには、次の 4 種類があります。

- **よく使用される名前** — その地域でよく使用される通り名です。通常は、通り上の特定の範囲の住所のみに対して使用されます。
- **省略形** — 通り名の省略形です。AddressLine1 の長さが 31 文字以上になる場合に使用することができます。例えば、1234 BERKSHIRE VALLEY RD APT 312A という通り名は、1234 BERKSHIRE VLLY RD APT 312A と省略することができます。
- **変更名** — 通り名が正式に変更された場合に、新しい名前を表すエイリアスです。例えば、SHINGLE BROOK RD という通り名が CANNING DR に変更された場合、CANNING DR が変更済みのエイリアス タイプとなります。
- **その他の名前** — この通り名エイリアスには、通りの他の名前や、通りの一般的な省略形などがあります。

エイリアスではない通り名のことを、基本通り名と呼びます。

入力において、よく使用されるエイリアスが使用されている場合は、このオプションを選択しているかどうかにかかわらず、そのエイリアスが出力の通り名になります。

これは、ValidateAddress でストリート名のエイリアスを処理する方法を制御する 3 つのオプションのうちの 1 つです。他の 2 つは OutputStreetNameAlias と OutputAbbreviatedAlias です。

多くの場合、OutputPreferredAlias と OutputAbbreviatedAlias の両方が選択されており、ValidateAddress が、郵便データベース内でよく使用されるエイリアスと省略形エイリアスの両方を検出した場合は、省略形エイリアスが出力に使用されます。入力の通り名がよく使用されるエイリアスである場合は、例外になります。この場合は、よく使用されるエイリアスが出力に使用されます。

- Y** 通り名に対してよく使用されるエイリアスの処理を実行します。
- N** 通り名に対してよく使用されるエイリアスの処理を実行しません。デフォルト

注：入力住所に、「変更名」である通り名のエイリアスが含まれている場合は、指定したオプションにかかわらず、出力住所には必ず、基本通り名が使用されます。

パラメータ	説明
OutputAbbreviatedAlias	<p>出力住所行の長さが 31 文字以上になる場合に、通り名に対する省略形エイリアスを出力に使用するかどうかを指定します。</p> <p>これは、<b>ValidateAddress</b> でストリート名のエイリアスを処理する方法を制御する 3 つのオプションのうちの 1 つです。他の 2 つは <b>OutputStreetNameAlias</b> と <b>OutputPreferredAlias</b> です。</p> <p>注: 入力において、よく使用されるエイリアスが指定されている場合は、通り名に対する省略形エイリアスの処理を有効にしても、出力の通り名は必ず、よく使用されるエイリアスになります。</p> <p><b>Y</b> 省略形エイリアスの処理を実行します。</p> <p><b>N</b> 省略形エイリアスの処理を実行しません。デフォルト</p> <p>注: 入力住所に、"変更名"である通り名のエイリアスが含まれている場合は、指定したオプションにかかわらず、出力住所には必ず、基本通り名が使用されます。</p>
DPVDetermineNoStat	<p>住所の "no stat" ステータスを調べます。住所が存在するが、郵便物を受け取れない場合、その住所は "no stat" とみなされるため、配達ルートに関する配達統計としてカウントされません (そのため "no stat" という用語が使用されます)。例としては、建設中の建物や、郵便物を受け取る可能性が低いと郵便配達業者が識別した建物などがあります。</p> <p><b>N</b> "no stat" ステータスを調べません。デフォルト</p> <p><b>Y</b> "no stat" ステータスを調べます。</p> <p>注: このオプションを使用するには DPV 処理を有効にする必要があります。</p> <p>結果は <b>DPVNoStat</b> フィールドに返されます。詳細については、<a href="#">LACSLink 出力 (648ページ)</a></p>
DPVDetermineVacancy	<p>そのロケーションがすくなくとも 90 日間使用されていないかどうかを調べます。</p> <p><b>N</b> 空家かどうかを調べません。デフォルト</p> <p><b>Y</b> 空家かどうかを調べます。</p> <p>注: このオプションを使用するには DPV 処理を有効にする必要があります。</p> <p>結果は <b>DPVVacant</b> フィールドに返されます。詳細については、<a href="#">LACSLink 出力 (648ページ)</a></p>



パラメータ	説明
ReturnVerimove	<p>出力に VeriMove 詳細データを返します。</p> <p><b>N</b> VeriMove 詳細データを返しません。デフォルト</p> <p><b>Y</b> VeriMove 詳細データを返します。</p>
SuppressZplusPhantomCarrierR777	<p>キャリアルート R777 の住所を抑制するかどうかを指定します。これらの住所は疑似ルートであり、通り配達に使用できません。これらの住所には USPS® による ZIP + 4® コードが割り当てられているため、Validate Address はこれらの住所を配達可能と判定します。キャリアルート R777 の住所を配達可能と判定したくない場合は、このオプションを選択します。その場合は、次のように動作します。</p> <ul style="list-style-type: none"> <li>• ZIP + 4 コードは割り当てられません</li> <li>• 住所は USPS Form 3553 (CASS Summary Report) から除外されます</li> <li>• DPV 補足コードとして R7 が返されます</li> </ul> <p><b>N</b> キャリアルート R777 の住所を抑制しません。</p> <p><b>Y</b> キャリアルート R777 の住所を抑制します。</p>
StreetMatchingStrictness	<p>入力住所が郵便データベース内の住所にマッチするかどうかを調べる際に使用するアルゴリズムを指定します。次のいずれかです。</p> <p><b>E</b> 入力された通り名は、データベースに完全に一致する必要があります。</p> <p><b>T</b> マッチングアルゴリズムは "厳格" です。</p> <p><b>M</b> マッチングアルゴリズムは "中" です (デフォルト)。</p> <p><b>L</b> マッチングアルゴリズムは "あいまい" です。</p>
FirmMatchingStrictness	<p>入力住所が郵便データベース内の住所にマッチするかどうかを調べる際に使用するアルゴリズムを指定します。次のいずれかです。</p> <p><b>E</b> 入力された企業名は、データベースに完全に一致する必要があります。</p> <p><b>T</b> マッチングアルゴリズムは "厳格" です。</p> <p><b>M</b> マッチングアルゴリズムは "中" です (デフォルト)。</p> <p><b>L</b> マッチングアルゴリズムは "あいまい" です。</p>

パラメータ	説明
DirectionalMatchingStrictness	<p>入力住所が郵便データベース内の住所にマッチするかどうかを調べる際に使用するアルゴリズムを指定します。次のいずれかです。</p> <p><b>E</b> 123 N Main St. における "N" など、入力された道順がデータベースに完全に一致する必要があります。</p> <p><b>T</b> マッチング アルゴリズムは "厳格" です。</p> <p><b>M</b> マッチング アルゴリズムは "中" です。デフォルト</p> <p><b>L</b> マッチング アルゴリズムは "あいまい" です。</p>
DPVSuccessfulStatusCondition	<p>DPV 結果がレコードの失敗の原因とならない一致条件を選択します。</p> <p><b>F</b> 全体一致</p> <p><b>P</b> 部分一致</p> <p><b>A</b> 常に一致。デフォルト</p> <p>注：このオプションを使用するには DPV 処理を有効にする必要があります。</p>
FailOnCMRAMatch	<p>民間私書箱（CMRA）との一致をマッチとみなしませんか。</p> <p><b>N</b> いいえ、CMRA との一致をマッチとみなします。デフォルト</p> <p><b>Y</b> はい、CMRA との一致をマッチとみなしません。</p> <p>注：このオプションを使用するには DPV 処理を有効にする必要があります。</p>
StandardAddressPMBLine	<p>私書箱 (PMB) の情報をどこに配置するかを指定します。</p> <p><b>N</b> 標準住所出力に PMB 情報を含めません (デフォルト)。</p> <p><b>1</b> PMB 情報を AddressLine1 に配置します。1 を指定した場合、StandardAddressFormat に C または D をセットする必要があります。</p> <p><b>2</b> PMB 情報を AddressLine2 に配置します。</p>

## パラメータ 説明

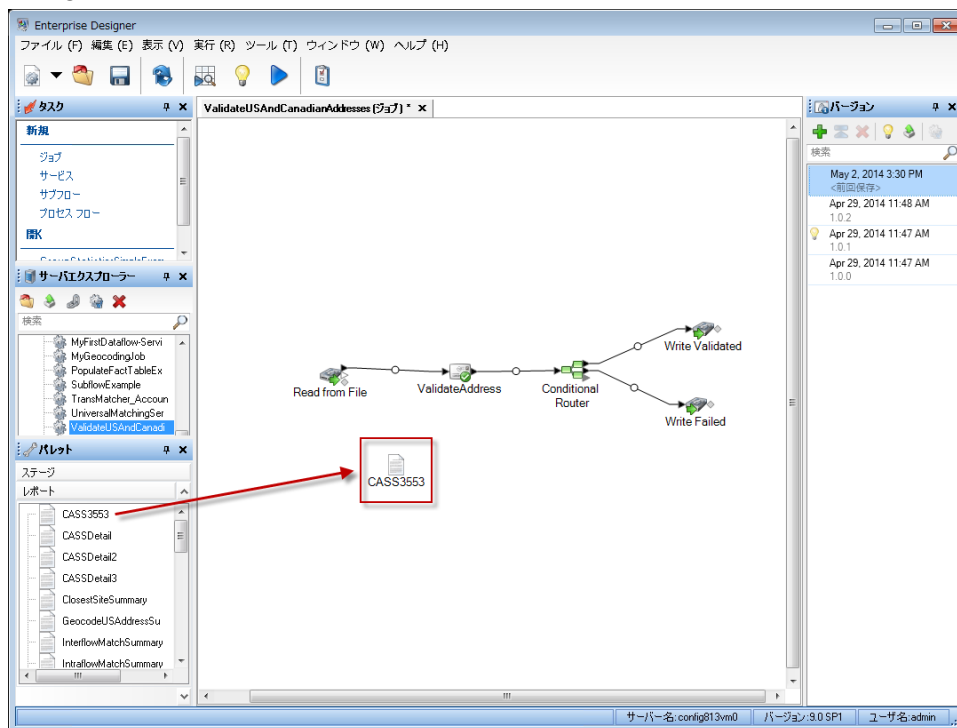
PreferredCity	<p>優先する最終行都市名を格納するかどうかを指定します。</p> <p><b>Z</b> USPS ZIP+4 ファイルからの Preferred Last Line City Name を格納します (都市名を上書き)。</p> <p>注：このオプションを選択すると、Validate Address は CASS 認定の設定と USPS 3553 レポートを生成します。</p> <p><b>C</b> USPS City/State ファイルからの USPS-preferred City Name を格納します</p> <p>注：このオプションを選択すると、Validate Address は CASS 認定の設定と USPS 3553 レポートを生成しません。</p> <p><b>P</b> USPS City/State ファイルからの Primary City Name を格納します</p> <p>注：このオプションを選択すると、Validate Address は CASS 認定の設定と USPS 3553 レポートを生成しません。</p>
---------------	---

## CASS 認定処理

また、CASS 認定™処理では USPS CASS 詳細レポートも生成されます。このレポートに含まれる情報は 3553 レポートと同じものですが、DPV、LACS、および SuiteLink に関する大幅に詳しい統計情報が含まれます。USPS CASS 詳細レポートは、郵便料金の値引きを受けるために必ずしも必要ではなく、郵便物と一緒に提出する必要はありません。

1. Validate Address を CASS 認定™モードにする必要があります。ウィンドウの一番上に **(CASS 認定でないもの)** と表示される場合は、**[CASS 有効]** ボタンをクリックしてください。**[CASS ルールを強制]** チェック ボックスが表示されます。
2. **[CASS 3553 を設定]** をクリックします。**[CASS レポート フィールド]** ダイアログ ボックスが表示されます。
3. **[リスト処理元]** の会社名、**[リスト名または ID 番号]**、およびこのジョブで処理する **[リスト数]** を入力します。
4. **[差出人名]**、**[住所]**、および **[都市]**、**[州]**、**[ZIP]** を入力します。
5. **[OK]** をクリックします。

生成された USPS® CASS Form 3553 のセクション B にリストの情報が、セクション D に差出人の情報が表示されます。

6. Enterprise Designer で、**CASS3553** レポートをレポートのパレットからキャンバスにドラッグ

グします。

7. キャンバスの **[CASS3553]** アイコンをダブルクリックします。
8. **[ステージ]** タブで、**[Validate Address]** チェックボックスをオンにします。Validate Address ステージを何か別の名前に変更している場合は、住所検証ステージに指定した名前のチェックボックスをオンにする必要があります。
9. **[パラメータ]** タブで、レポートのフォーマットを選択します。PDF、HTML、またはプレーンテキストのフォーマットでレポートを作成できます。
10. **[OK]** をクリックします。
11. CASS 詳細レポートを生成する場合は、**CASSDetail** に対して手順 6 ~ 10 を繰り返します。

カナダ住所のオプション

## パラメータ

## 説明

---

**PerformCanadianProcessing**

カナダ住所を処理するかどうかを指定します。カナダ住所処理を有効にした場合、**ValidateAddress** はカナダ住所の検証を試みます。カナダ住所処理を無効にすると、**Status** 出力フィールドに "F" が設定されて返され、カナダ住所処理は失敗します。その際、出力フィールド **Status.Code** は、"DisabledCoder" となります。カナダ住所処理のライセンスを取得していない場合は、ジョブにカナダ住所が含まれるか否かにかかわらず、カナダ住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：カナダ住所を正常に処理するには、カナダ住所処理の有効なライセンスを取得する必要があります。カナダ住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、カナダ住所処理を有効にすると、エラーとなります。

- N**      カナダ住所を処理しません。
- Y**      カナダ住所を処理します (デフォルト)。

---

**Database.Canada**

カナダ住所の検証に使用するデータベースを指定します。カナダ住所検証用のデータベースを指定するには、**[データベース]** ドロップダウンリストからデータベースを選択します。**Management Console** の **[カナダ データベース リソース]** パネルで定義されたデータベースのみが使用可能です。

---

## パラメータ

## 説明

## CanFrenchFormat

住所及び方向指示の書式設定に使用する言語 (英語またはフランス語) の判断方法を指定します。以下に、英語およびフランス語で書式設定された住所の例を示します。

英語: 123 Main St W フランス語: 123 Rue Main O

このパラメータは、住所の書式設定を制御します。また、方向指示の綴りにも影響を与えますが、接尾語の綴りには影響を与えません。

- C** マッチング処理によって返される通り接尾語によって、言語を判断します。マッチング処理によって返されるストリート接尾語は、`ValidateAddress` が処理において内部で使用するもので、入力住所のものとは異なる場合があります。あいまいなレコードは、入力と同様に書式設定されます。デフォルトケベック州の住所はすべて、フランス語で書式設定されます。
- S** カナダ データベースによって、言語を判断します。カナダ データベースには、**Canada Post Corporation** (CPC: カナダ郵政公社) からのデータが含まれています。ケベック州の住所はすべて、フランス語で書式設定されます。
- T** `CanLanguage` 入力フィールドによって、言語を判断します。このフィールドに空白以外の値が設定されている場合は、住所はフランス語で書式設定されます。

## CanEnglishApartmentLabel

英語の住所に対し、入力住所にアパートメント ラベルが存在しない場合に、出力に使用するデフォルト アpartment ラベルを指定します。

`CanStandardAddressFormat=F` と指定した場合、この設定は無視されます。

- Apt** ラベルとして "Apt" を使用します。デフォルト
- Apartment** ラベルとして "Apartment" を使用します。
- Suite** ラベルとして "Suite" を使用します。
- Unit** ラベルとして "Unit" を使用します。

## パラメータ

## 説明

## CanFrenchApartmentLabel

フランス語の住所に対し、入力住所にアパートメント ラベルが存在しない場合に、出力に使用するデフォルト アパートメント ラベルを指定します。  
CanStandardAddressFormat=F と指定した場合、この設定は無視されます。

<b>App</b>	"App" をラベルとして使用します。デフォルト
<b>Appartement</b>	Use "Appartement" as the label.
<b>Bureau</b>	Use "Bureau" as the label.
<b>Suite</b>	Use "Suite" as the label.
<b>Unite</b>	Use "Unite" as the label.

## ForceCorrectionLVR

正式情報やスイート情報を変更して、Large Volume Receiver (LVR) または Single-Single レコードをマッチさせます (その郵便番号/通り名/通り タイプに対して 1 つしかレコードがない場合に使用します)。

- N** LVR または Single-Single レコードをマッチさせるために正式情報やスイート情報を変更しません。LVR レコードは、有効だが修正不可能なレコード (VN) としてマーク付けされます。Single-Single レコードは可能ならば修正されます。または、修正不可能なレコードとして処理されます。
- Y** LVR または Single-Single レコードをマッチさせるために正式情報やスイート情報を変更します。

注：このチェック ボックスをオンにする場合は、SERP 認定の設定 ではないため、Statement of Address Accuracy は印刷されません。

## パラメータ

## 説明

---

**CanPreferHouseNum**

家番号と郵便番号がともに有効であるが、競合する場合、CanPreferHouseNum=Y と指定することによって、家番号に合わせて郵便番号を強制的に修正できます。このオプションを選択しない場合、郵便番号に合わせて家番号が変更されます。

- N** 郵便番号に合わせて家番号を変更します。デフォルト
- Y** 家番号に合わせて郵便番号を変更します。

---

**CanOutputCityAlias**

入力住所に都市名のエイリアスがある場合、そのエイリアスを返すかどうかを指定します。CanOutputCityFormat=Dを指定している場合は、このオプションは無効です。

- Y** 入力に都市名のエイリアスがある場合は、都市名のエイリアスを出力します。デフォルト
- N** 入力に都市名のエイリアスがあっても、都市名のエイリアスを出力しません。

---

**CanNonCivicFormat**

出力において、正式住所ではないキーワードを短縮するかどうかを指定します。例えば、Post Office Box と PO Box のどちらを使用するかが決まります。

- A** 正式住所ではないキーワードを短縮します。デフォルト
- F** 正式住所ではないキーワードを短縮しません。正式なキーワードを使用します。

---

**EnableSERP**

SERP オプションを使用するかどうかを指定します。

- Y** SERP オプションを有効にします。
  - N** SERP オプションを有効にしません。デフォルト
-



## パラメータ

## 説明

---

**CanStandardAddressFormat**

出力住所における、補助的な住所情報の配置場所を指定します。補助的な住所情報とは、部屋番号やアパート番号などの指定子のことです。

- D** アパート情報を、で指定された場所に配置します。  
デフォルト
- B** アパート情報を、**AddressLine1** フィールドの末尾に配置します。
- F** アパート番号のみ(ラベルは除く)を、**AddressLine1** フィールドの先頭に配置します。例えば、**400-123 Rue Main** とします。
- E** アパート番号とラベルを、**AddressLine1** フィールドの先頭に配置します。例えば、**Apt 400 123 Rue Main** とします。
- S** アパート情報を別の行に配置します。
- S** アパート情報を入力住所と同じ場所に配置します。

---

**CanOutputCityFormat**

都市の名前が長い場合に、**long**、**medium**、もしくは **short** のどの都市名を使用するかを指定します。例を次に示します。

Long: BUFFALO HEAD PRAIRIE

Medium: BUFFALO-HEAD-PR

Short: BUFFALO-HD-PR

- D** **OutputShortCityName** オプションで指定されたデフォルトオプションを使用します。デフォルト **OutputShortCityName=V** を指定した場合、都市は、このオプションで **V** を選択し (以下を参照)、**CanOutputCityAlias** で **Y** を選択した場合と同じように書式設定されます。
  - S** 短い都市名を出力します。
  - L** 長い都市名を出力します。
  - M** 中間の長さの都市名を出力します。
  - I** 入力住所と同じ都市フォーマットを使用します。出力は、**L**、**M**、または **S** です。
-

## パラメータ

## 説明

---

**CanRuralRouteFormat**

地方配送路の配達情報を配置する場所を指定します。地方配送路の配達情報を含む住所の例を以下に示します。

**36 GRANT RD RR 3ANTIGONISH NS**

この住所において、"RR 3"は地方配送路の配達情報です。

- A** 地方配送路の配達情報を、住所と同一行の住所情報の後に配置します。デフォルト例を次に示します。

**36 GRANT RD RR 3**

- S** 地方配送路の配達情報を、別の住所行に配置します。例を次に示します。

**36 GRANT RDRR 3**

---

**CanDeliveryOfficeFormat**

配達局情報の配置場所を指定します。配達局情報を含む住所の例を次に示します。

**PO BOX 8625 STN AST.JOHN'S NL**

- I** 配達局情報を、入力住所と同じ場所に配置します。デフォルト

- A** 配達局情報を、住所と同一行の住所情報の後に配置します。例を次に示します。

**PO BOX 8625 STN A**

- S** 配達局情報を、別の住所行に配置します。例を次に示します。

**PO BOX 8625STN A**

---

## パラメータ

## 説明

## CanDualAddressLogic

住所に、正式情報と非正式情報の両方が含まれている場合に、**ValidateAddress** がストリート一致と、**PO Box**/非正式一致のどちらを返すかを指定します。次のいずれかです。

- D** **DualAddressLogic** のグローバル オプションを使用します。デフォルト
- P** **PO Box** など通り以外のデータとマッチングします。
- S** 通りとマッチングします。

例えば、次の入力住所が与えられたとします。

AddressLine1: 36 GRANT RD  
AddressLine2: RR 4  
City: ANTIGONISH  
StateProvince: NS

**ValidateAddress** は、次のいずれかを返します。

- **CanDualAddressLogic** が **S** に設定されている場合、**ValidateAddress** は次を返します。

AddressLine1: 36 GRANT RD  
AddressLine2: RR 3  
City: ANTIGONISH  
StateProvince: NS  
PostalCode: B2G 2L1

- **CanDualAddressLogic** が **P** に設定されている場合、**ValidateAddress** は次を返します。

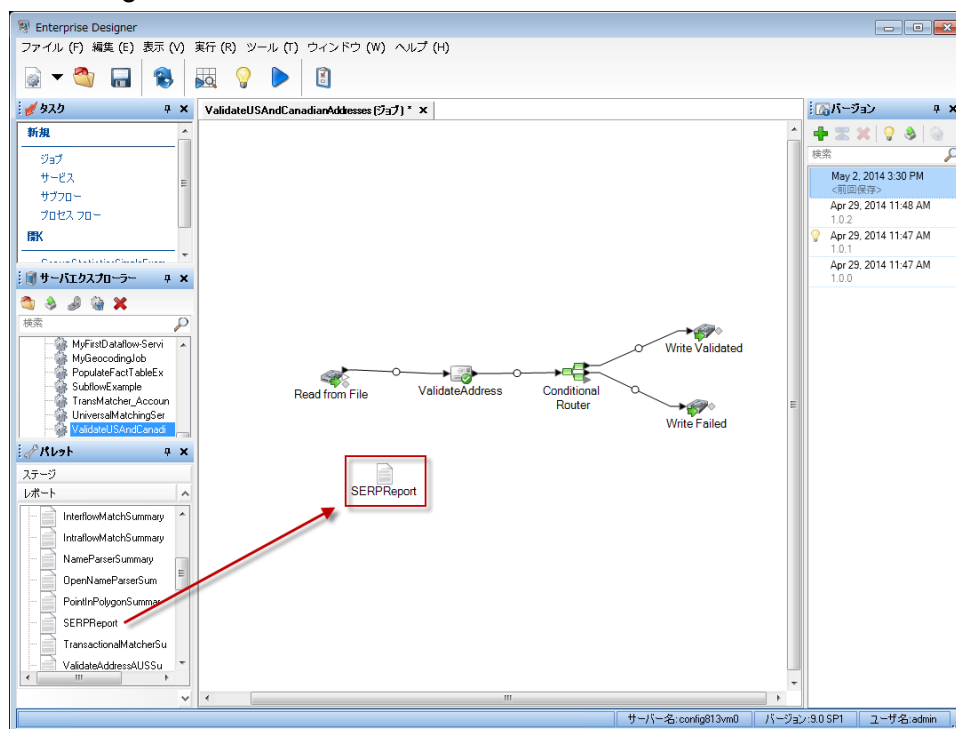
AddressLine1: RR 4  
City: ANTIGONISH  
StateProvince: NS  
PostalCode: B2G 2L2

住所の正規化に使用されない住所データは、**AdditionalInputData** フィールドに返されます。詳細については、「[出力データオプション \(573ページ\)](#)」を参照してください。

## SERP の処理

1. **Validate Address** を SERP 認定™モードにする必要があります。ウィンドウの一番上に (SERP 認定でないもの) と表示される場合は、**[SERP 設定を有効にする]** ボタンをクリックしてください。**[SERP を設定]** ボックスが表示されます。
2. **[SERP を設定]** をクリックします。**[SERP レポート フィールド]** ダイアログ ボックスが表示されます。
3. 荷主の **[CPC 番号]** を入力します。
4. 差出人の **[名前]**、**[住所]**、および **[都市]**、**[州]**、**[郵便番号]** を入力します。

5. **[OK]** をクリックします。
6. Enterprise Designer で、SERP レポートをレポートのパレットからキャンバスにドラッグし



ます。

7. キャンバスの **[SERPReport]** アイコンをダブルクリックします。
8. **[ステージ]** タブで、**[Validate Address]** チェックボックスをオンにします。Validate Address ステージを何か別の名前に変更している場合は、住所検証ステージに指定した名前のチェックボックスをオンにする必要があります。
9. **[パラメータ]** タブで、レポートのフォーマットを選択します。PDF、HTML、またはプレーンテキストのフォーマットでレポートを作成できます。デフォルトではPDFフォーマットで作成されます。
10. **[OK]** をクリックします。

#### SERP リターン コードの取得

SERP リターン コードは、カナダ郵政公社の Software Evaluation and Recognition Program の規定によって定められる入力住所の品質を表します。

SERP リターン コードを取得するには、OutputRecordType=P を指定します。出力データオプション (573ページ) OutputRecordTypeの詳細については、を参照してください。

SERP リターン コードは、以下の出力フィールドに返されます。

表 129 : SERP リターン コード出力

応答要素	説明
CanadianSERPCode	<p data-bbox="557 443 1032 474">検証/修正リターン コード (カナダ住所のみ)。</p> <p data-bbox="557 491 1403 558"><b>V</b> 入力は無効です。カナダ郵政公社は、以下のすべての条件を満たす住所を、「有効な」住所であると定義しています。</p> <p data-bbox="691 575 1403 642">注：一部例外があります。詳細については、CPC にお問い合わせください。</p> <ul data-bbox="607 680 1421 1050" style="list-style-type: none"> <li>• 住所は、CPC の Postal Code Data Files に示されるとおりに、すべての必須コンポーネントを含む必要があります。</li> <li>• 住所は、CPC の Postal Code Data Files の単一の住所のみに対し、すべてのコンポーネントにおいて完全に一致する必要があります。ただし、CPC Postal Code Data Files に示されている、許容される別の語および名前が使用されていてもかまいません。</li> <li>• 住所コンポーネントは、あいまいな部分がなく、はっきりと認識できる形式である必要があります。一部のコンポーネントには、それらを識別するための "修飾子" が必要な場合があります。例えば、Route Service の住所には、同じ番号の "Suburban Service" または "SS" の住所と区別するために、"Rural Route" または "RR" のキーワードが必要です。</li> </ul> <p data-bbox="557 1079 1421 1180"><b>I</b> 入力は無効です。「無効な」住所とは、有効な住所に対する CPC の条件を満たさない住所のことです (上記を参照)。例としては、住所コンポーネントが欠落している、無効である、または矛盾が存在する場合が挙げられます。</p> <p data-bbox="557 1205 1421 1272"><b>C</b> 入力は修正可能です。「修正可能な」住所とは、修正することによって、単一の住所のみに一致させることのできる住所のことです。</p> <p data-bbox="557 1289 1421 1390"><b>N</b> 入力は修正不可能です。「修正不可能な」住所とは、複数の異なる修正方法があり得るために ValidateAddress が単一の修正住所を特定できない住所のことです。</p> <p data-bbox="557 1415 1052 1444"><b>F</b> 入力住所は外国 (カナダ以外) の住所です。</p>

## 国際住所オプション

米国とカナダ以外の住所は "国際" 住所と呼ばれます。以下に、国際住所の処理をコントロールするオプションについて説明します。

## パラメータ

## 説明

---

**PerformInternationalProcessing**

国際住所(米国およびカナダ以外の住所)を処理するかどうかを指定します。国際住所処理を有効にした場合、**ValidateAddress** は国際住所の妥当性を確認します。国際住所処理を無効にした場合、**Status**フィールドに "F" が設定され国際住所は失敗します。出力フィールド **Status.Code** は、"DisabledCoder" となります。国際住所処理のライセンスを取得していない場合は、ジョブに国際住所が含まれるか否かにかかわらず、国際住所処理を無効にしなければ、ジョブを正常に実行することはできません。

注：国際住所を正常に処理するには、国際住所処理の有効なライセンスを取得する必要があります。国際住所処理のライセンスを取得していないか、ライセンスの期限が切れているにもかかわらず、国際住所処理を有効にすると、エラーとなります。

**N** 国際住所を処理しません。

**Y** 国際住所を処理します (デフォルト)。

---

**Database.International**

国際住所の妥当性の確認に使用するデータベースを指定します。国際住所検証用のデータベースを指定するには、**【データベース】**ドロップダウンリストからデータベースを選択します。**Management Console** の **【国際データベースリソース】** パネルで定義されたデータベースのみが使用可能です。

---

## パラメータ

## 説明

---

**InternationalCityStreetSearching**

デフォルトでは、**ValidateAddress** は、住所マッチングの精度とパフォーマンスのバランスをうまくとります。マッチング精度を犠牲にしてパフォーマンスを向上させる場合は、**InternationalCityStreetSearching** オプションを使用して、処理速度を上げます。これを実行すると、精度はやや低下します。このオプションは、米国およびカナダ以外の住所のパフォーマンスのみを制御します。この設定が影響を与えるレコードの割合は少なく、大部分が英国の住所です。米国およびカナダ住所処理のパフォーマンスは制御できません。

**GetCandidateAddresses** を使用した場合に

**GetCandidateAddresses** が返す候補住所は、国際住所のパフォーマンス チューニング オプションを 100 以外の任意の値に設定した場合に **ValidateAddress** が返す複数マッチと異なることがあります。

パフォーマンスを制御するには、0 ~ 100 の値を指定してください。100 を設定すると精度が最大化し、0 を設定すると速度が最大化します。デフォルト値は 100 です。

---

**AddressLineSearchOnFail**

このオプションにより、**ValidateAddress** において、**City**、**StateProvince**、および **PostalCode** の各入力フィールドの値を使用して住所にマッチする結果が得られなかった場合に、**AddressLine** 入力フィールドで都市、州/省、郵便番号、および国を検索することができます。

入力住所において、**AddressLine** フィールドに都市、州/省、および郵便番号の情報が存在する場合は、このオプションを有効にすることを検討してください。

入力住所において、**City**、**State/Province**、および **PostalCode** フィールドが使用されている場合は、このオプションを無効にしてください。このオプションを有効にしてこれらのフィールドを使用すると、**ValidateAddress** がこれらのフィールド値の修正 (例えば、スペルミスのある都市名など) に失敗する可能性が高くなります。

- N** いいえ、**AddressLine** フィールドを検索しません。
- Y** はい、**AddressLine** フィールドを検索します。こちらがデフォルトです。

## 応答

ValidateAddress からの出力には、選択した出力カテゴリに応じて異なる情報が含まれます。

### 標準住所出力

標準住所出力は、宛名ラベルに表記される住所に対応する 4 行の住所で構成されます。都市、州/省、郵便番号などのデータも、標準住所出力に含まれます。OutputRecordType=A と設定した場合、妥当性を確認した住所に対し、標準住所出力が返されます。妥当性が確認できなかった住所に対しては、標準住所フィールドが必ず返されます。妥当性が確認されなかった住所に対しては、標準住所出力フィールドには、入力住所がそのまま含まれます ("パス スルー" データ)。妥当性が確認できなかった場合に、郵便当局の規格に従って住所を正規化するには、リクエスト時に OutputFormattedOnFail=Y を指定します。

表 130 : 標準住所出力

応答要素	説明
AdditionalInputData	住所検証プロセスで使用されない入力データ。詳細については、「 <a href="#">AdditionalInputData について (654 ページ)</a> 」を参照してください。
AddressLine1	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 1 行目です。住所の妥当性が確認できなかった場合は、入力住所の 1 行目がそのまま出力されます。
AddressLine2	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 2 行目です。住所の妥当性が確認できなかった場合は、入力住所の 2 行目がそのまま出力されます。
AddressLine3	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 3 行目です。住所の妥当性が確認できなかった場合は、入力住所の 3 行目がそのまま出力されます。
AddressLine4	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 4 行目です。住所の妥当性が確認できなかった場合は、入力住所の 4 行目がそのまま出力されます。



応答要素	説明
AddressLine5	英国住所にのみ適用されます。住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の5行目です。住所の妥当性が確認できなかった場合は、入力住所の5行目がそのまま出力されます。
City	妥当性が確認された都市名。
Country	OutputCountryFormat で選択した、以下のいずれかのフォーマットで示された国。 <ul style="list-style-type: none"><li>• ISO コード</li><li>• UPU コード</li><li>• 英語</li><li>• フランス語</li><li>• ドイツ語</li><li>• スペイン語</li></ul>
DepartmentName	企業内の下位区分(英国住所においてのみ)。例えば、エンジニアリング部門などです。
FirmName	妥当性が確認された企業名。
PostalCode	妥当性が確認された ZIP Code™ または郵便番号。
PostalCode.AddOn	ZIP Code™ の4桁のアドオン部分。例えば、60655-1844 という ZIP Code™ において、4桁のアドオン部分は 1844 になります(米国住所のみ)。
PostalCode.Base	5桁の ZIP Code™。例えば、20706 (米国住所のみ)。
StateProvince	妥当性が確認された州または省の略称。
USUrbanName	妥当性が確認された都市の都市化名。(米国住所のみ)。主にプエルトリコの住所で使用されます。

## パース済み住所要素出力

OutputRecordType=E を設定した場合、出力住所は、パース済み住所の形式で書式設定されます。妥当性が確認できなかった場合に、パース済み住所形式で書式設定されたデータ(正規化済み住所)を返すには、OutputFormattedOnFail=Y を指定します。

注：妥当性が確認できたかどうかにかかわらず、常にパースした入力データを返すには、OutputRecordType=I を指定します。詳細については、「[パース済み入力 \(700ページ\)](#)」を参照してください。

表 131 : パース済み住所出力

応答要素	説明
AdditionalInputData	ValidateAddress で使用されない入力データ。詳細については、「 <a href="#">AdditionalInputData について (654ページ)</a> 」を参照してください。
AdditionalInputData.Base	ValidateAddress によって正規化済み住所に出力されなかった入力データ。詳細については、「 <a href="#">AdditionalInputData について (654ページ)</a> 」を参照してください。
AdditionalInputData.Unmatched	マッチャーに引き渡されたが、ValidateAddress による検証に使用されなかった入力データ。詳細については、「 <a href="#">AdditionalInputData について (654ページ)</a> 」を参照してください。
ApartmentLabel	アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b>
ApartmentLabel2	補助的なアパート指定子。例: 123 E Main St Apt 3, 4th <b>Floor</b>  注: このリリースでは、このフィールドは常に空白になります。
ApartmentNumber	アパート番号。例: 123 E Main St <b>APT 3</b>

応答要素	説明
ApartmentNumber2	補助的なアパート番号。例: 123 E Main St APT 3, <b>4th</b> Floor 注: このリリースでは、このフィールドは常に空白になります。
CanadianDeliveryInstallationAreaName	配達施設名 (カナダ住所のみ)
CanadianDeliveryInstallationQualifierName	配達施設の修飾子 (カナダ住所のみ)
CanadianDeliveryInstallationType	配達施設の種類 (カナダ住所のみ)
City	妥当性が確認された都市名
Country	国。フォーマットは、OutputCountryFormat で選択したものになります。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> <li>• フランス語</li> <li>• ドイツ語</li> <li>• スペイン語</li> </ul>
DepartmentName	企業内の下位区分(英国住所においてのみ)。例えば、エンジニアリング部門などです。
FirmName	妥当性が確認された企業名
HouseNumber	家番号 1。例: <b>123</b> E Main St Apt 3
LeadingDirectional	接頭方向指示。例: <b>123 E</b> Main St Apt 3

応答要素	説明
POBox	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode	妥当性が検証された郵便番号。米国式の住所の場合に、ZIP Code と呼びます。
PrivateMailbox	私設私書箱のインジケータ。
PrivateMailbox.Type	私設私書箱の種別。次のような値があります。 <ul style="list-style-type: none"> <li>• Standard</li> <li>• Non-Standard</li> </ul> <p>注：PrivateMailboxType (フィールド名にピリオドなし) に換わるものです。これに従って API 呼び出しを修正してください。</p>
RRHC	地方配送路/幹線請負契約のインジケータ
StateProvince	妥当性が確認された州または省の名前
StreetName	通り名。例: 123 E Main St Apt 3
StreetSuffix	通り接尾語。例: 123 E Main St Apt 3
TrailingDirectional	接尾方向指示。例: 123 Pennsylvania Ave NW
USUrbanName	USPS® 都市化名。プエルトリコ住所のみ。

### パース済み入力

出力には、パース済み形式で入力住所を含めることができます。このようなタイプの出力は、"パース済み入力" と呼ばれます。パース済み入力フィールドには、`ValidateAddress` が住所の妥当性を

検証したかどうかにかかわらず、入力として使用される住所データが含まれます。パース済み入力は、住所の妥当性を検証できた場合にパース済み住所要素に妥当性が検証された住所が含まれ、オプションで、住所の妥当性が検証できなかった場合には入力データが含まれるという点で、"パース済み住所要素" 出力と異なります。パース済み入力には、`ValidateAddress` が住所の妥当性を検証したかどうかにかかわらず、常に入力住所が含まれます。

パース済み入力フィールドを出力に含めるには、`OutputRecordType=I` を設定します。

表 132 : パース済み入力

応答要素	説明
<code>ApartmentLabel.Input</code>	アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b>
<code>ApartmentNumber.Input</code>	アパート番号。例: 123 E Main St <b>APT 3</b>
<code>CanadianDeliveryInstallationAreaName.Input</code>	配達施設名 (カナダ住所のみ)
<code>CanadianDeliveryInstallationQualifierName.Input</code>	配達施設の修飾子 (カナダ住所のみ)
<code>CanadianDeliveryInstallationType.Input</code>	配達施設の種類 (カナダ住所のみ)
<code>City.Input</code>	妥当性が確認された都市名
<code>Country.Input</code>	<p>国。フォーマットは、<code>OutputCountryFormat</code> で選択したものになります。</p> <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> <li>• フランス語</li> <li>• ドイツ語</li> <li>• スペイン語</li> </ul>
<code>FirmName.Input</code>	妥当性が確認された企業名

応答要素	説明
HouseNumber.Input	家番号 1。例: <b>123</b> E Main St Apt 3
LeadingDirectional.Input	接頭方向指示。例: 123 <b>E</b> Main St Apt 3
POBox.Input	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode.Input	妥当性が検証された郵便番号。米国住所に対しては、ZIP Code と呼びます。
PrivateMailbox.Input	私設私書箱のインジケータ
PrivateMailbox.Type.Input	私設私書箱の種別。次のような値があります。 <ul style="list-style-type: none"><li>標準</li><li>非標準</li></ul>
RRHC.Input	地方配送路/幹線請負契約のインジケータ
StateProvince.Input	妥当性が確認された州または省の名前
StreetName.Input	通り名。例: 123 <b>E Main St</b> Apt 3
StreetSuffix.Input	通り接尾語。例: 123 E Main St Apt 3
TrailingDirectional.Input	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>
USUrbanName.Input	USPS® 都市化名

## 郵便データ出力

OutputRecordType に P が含まれる場合、以下のフィールドが出力として返されます。

表 133 : 郵便データ出力

応答要素	説明
CanadianSERPCode	検証/修正リターン コード (カナダ住所のみ)。詳細については、「 <a href="#">SERP リターンコードの取得 (612ページ)</a> 」を参照してください。
IntHexaviaCode	通りを表す数値コード (フランスの住所の場合のみ)。Hexavia コードの詳細については、 <a href="http://www.laposte.fr">www.laposte.fr</a> を参照してください。
IntINSEECODE	都市を表す数値コード (フランスの住所の場合のみ)。INSEE コードの一覧については、 <a href="http://www.insee.fr">www.insee.fr</a> を参照してください。
PostalBarCode	配達ポイント バーコードの 2 桁の配達ポイント部分 (米国住所のみ)。詳細については、「 <a href="#">配達ポイント バーコードの作成 (582ページ)</a> 」を参照してください。
USAltAddr	他の住所マッチング ロジックを使用したかどうか、使用した場合はどのロジックを使用したかを表します (米国住所のみ)。次のいずれかです。 <ul style="list-style-type: none"> <li><b>NULL</b>      他の住所スキームを使用していません。</li> <li><b>D</b>            別の配達ポイント ロジックを使用しました。</li> <li><b>E</b>            別の高層マッチ ロジックを使用しました。</li> <li><b>S</b>            小都市デフォルト ロジックを使用しました。</li> <li><b>U</b>            ユニークな ZIP Code ロジックを使用しました。</li> </ul>
USBCCheckDigit	11 桁の配達ポイント バーコードのチェック デジット部分 (米国住所のみ)。詳細については、「 <a href="#">配達ポイントバーコードの作成 (582ページ)</a> 」を参照してください。
USCarrierRouteCode	配達ルート コード (米国住所のみ)。詳細については、「 <a href="#">配達ルート コードの取得 (582ページ)</a> 」を参照してください。

応答要素	説明
USCongressionalDistrict	下院選挙区出力 (米国住所のみ)。詳細については、「 <a href="#">下院選挙区の取得</a> (580ページ)」を参照してください。
USCountyName	郡名出力 (米国住所のみ)。詳細については、「 <a href="#">郡名の取得</a> (581ページ)」を参照してください。
USFinanceNumber	住所の所在地の Finance Number (米国住所のみ)。Finance Number とは、複数の ZIP Code を含む地域に USPS が割り当てた番号です。住所の Finance Number が米国データベースの候補住所の Finance Number に一致した場合のみ、住所の妥当性確認に成功します。
USFIPSCountyNumber	FIPS (連邦情報処理標準) 郡番号 (米国住所のみ)。詳細については、「 <a href="#">FIPS 郡番号の取得</a> (581ページ)」を参照してください。
USLACS	住所が、LACS <sup>Link</sup> 変換の候補であるかどうかを表します (米国住所のみ)。次のいずれかです。 <ul style="list-style-type: none"> <li><b>Y</b> 住所は LACS<sup>Link</sup> 処理の候補です。LACS<sup>Link</sup> が有効である場合、LACS<sup>Link</sup> データベースを使用して住所を変換しようとします。変換に成功した場合、出力住所は LACS<sup>Link</sup> データベースから取得した新しい住所になります。変換できなかった場合は、住所は変換されません。</li> <li><b>N</b> 住所は LACS<sup>Link</sup> 処理の候補ではありません。ただし、LACS<sup>Link</sup> 処理が要求され、LACS<sup>Link</sup> データベースがインストールされており、かつ、次の条件のいずれかが満たされている場合は、LACS<sup>Link</sup> 処理が行われる場合があります。 <ul style="list-style-type: none"> <li>• 住所が地方配送路住所にマッチし、RecordType.Default フィールドで Y が返された場合。</li> <li>• 入力住所が、米国郵便データベースのいずれの住所にもマッチしなかった場合 (複数にマッチしたことによる失敗は、LACS<sup>Link</sup> の候補にはなりません)。</li> </ul> </li> </ul>
USLastLineNumber	主要都市が同一である複数の ZIP Code を同一グループにまとめる 6 文字の英数字の値。例えば、最終行が次の 2 つのいずれかである住所は、最終行番号が同一になります。 <p>Chantilly VA 20151</p> <p>Chantilly VA 20152</p>



## 結果インジケータ

結果インジケータは、住所に対して実行した処理の種類に関する情報を提供します。結果インジケータには、次の 2 種類があります。

### レコード レベルの結果インジケータ

レコード レベルの結果インジケータは、各レコードに対する `ValidateAddress` 処理の結果に関するデータを提供します。例えば、マッチングの成功または失敗、住所を処理したコーダーなどの詳細情報を示します。以下の表に、`ValidateAddress` が返すレコード レベルの結果インジケータの一覧を示します。

表 134 : レコード レベル インジケータ

応答要素	説明
AddressFormat	返された住所データのタイプ。
	<b>F</b> フランス語フォーマット (例: 123 Rue Main)
	<b>E</b> 英語フォーマット (例: 123 Main St)
Confidence	返された住所に割り当てられた確信レベル。範囲は 0 ~ 100 です。0 は失敗を表し、100 はマッチ結果が正しいことに対する確信レベルが非常に高いことを表します。複数のマッチ結果がある場合、確信レベルは 0 です。この数値の計算方法については、 <a href="#">Validate Address 確信アルゴリズムの概要 (762ページ)</a> を参照してください。
CouldNotValidate	<p>マッチしなかった場合の、妥当性が確認できなかった住所コンポーネント。</p> <ul style="list-style-type: none"> <li>• ApartmentNumber</li> <li>• HouseNumber</li> <li>• StreetName</li> <li>• PostalCode</li> <li>• City</li> <li>• Directional</li> <li>• StreetSuffix</li> <li>• Firm</li> <li>• POBoxNumber</li> <li>• RuralRoute</li> </ul> <p>注：複数のコンポーネントがカンマ区切りリストとして返されることがあります。</p>

## 応答要素

## 説明

## CountryLevel

利用可能な住所マッチングのカテゴリ。米国およびカナダの住所に対しては、常に「A」です。次のいずれかです。

- A** 住所は非常に詳細な郵便データを利用できる国にあります。このマッチレベルにある住所では、以下の住所要素を検証および修正でき、入力から欠落している場合は追加できます。
- 郵便番号
  - 都市名
  - 州/郡名
  - 通り住所要素
  - 国名
- B** 住所は中程度の詳細さの郵便データを利用できる国にあります。このマッチレベルにある住所では、以下の住所要素を検証および修正でき、入力から欠落している場合は追加できます。
- 郵便番号
  - 都市名
  - 州/郡名
  - 国名
- C** 住所は郵便データが詳細ではない国にあります。このマッチレベルにある住所に対して、以下のアクションを実行することができます。
- 国名の検証および修正 (欠落している国名を補うことはできません)
  - 郵便番号のフォーマットの検証 (欠落している郵便番号を補ったり、番号を検証することはできません)

## 応答要素

## 説明

---

**MatchScore**

**MatchScore** は、出力住所がどの程度正しいかを示します。**MatchScore** は、マッチ結果を得るために入力住所をどれだけ変更したかを表す **Confidence** とはまったく異なるものです。**MatchScore** の意味は、米国住所と米国以外の住所で異なります。

米国住所に対しては、**MatchScore** は 0 ~ 9 の段階に対応する 1 桁のスコアで、(**ValidateAddress** による変換があればその実行後に) 通り名一致の近接性を反映します。0 は完全一致を意味し、9 は最も可能性の低い一致を意味します。マッチしなかった場合、このフィールドは空白です。

米国とカナダ以外の住所では、**MatchScore** は 5 桁のスコアで、最大値は 00999 です。数字が大きいほど、より近い一致を意味します。

このフィールドは、カナダの住所には適用されません。

米国住所のマッチ スコアと米国以外の住所のマッチ スコアは、同等と見なすことはできないことに注意してください。例えば、米国住所に対するマッチスコア 4 は、米国以外の住所に対する 00004 と同じマッチ レベルを意味するものではありません。

注： **Validate Address** および **Advanced Matching** モジュールのコンポーネントは、どちらも **MatchScore** フィールドを使用します。データフローの出力の **MatchScore** フィールドの値は、出力ステージに送られる前に最後に値を変更したステージによって決まります。データフローに **Validate Address** および **Advanced Matching** モジュールのコンポーネントが含まれ、各ステージの **MatchScore** 出力フィールドを確認したい場合は、**Transformer** ステージを使用して、**MatchScore** 値を他のフィールドにコピーしてください。例えば、**Validate Address** によって **MatchScore** という出力フィールドが作成され、**Transformer** ステージによって **Validate Address** の **MatchScore** フィールドが **AddressMatchScore** というフィールドにコピーされます。マッチャー ステージを実行すると、マッチャーから得た値が **MatchScore** フィールドに設定され、**Validate Address** から得た **AddressMatchScore** の値が引き渡されます。

---

**MultimatchCount**

複数のマッチが検出された場合、一致する可能性のあるレコードの数を示します。

---

応答要素	説明
MultipleMatches	<p>複数のマッチが検出された場合に、複数のマッチを持つ次の住所コンポーネントを示します。</p> <ul style="list-style-type: none"> <li>• Firm</li> <li>• LeadingDirectional</li> <li>• PostalCode</li> <li>• StreetName</li> <li>• StreetSuffix</li> <li>• TrailingDirectional</li> <li>• Urbanization</li> </ul> <p>注：複数のコンポーネントがカンマ区切りリストとして返されることがあります。</p>
ProbableCorrectness	<p>一致の精度を 0 ~ 9 の段階で示します。結果は次のようになります。</p> <ul style="list-style-type: none"> <li>• 空白 - 一致するものが見つかりませんでした</li> <li>• 0 - 正確である可能性が非常に高い (完全一致)</li> <li>• 1-8 - 正確である可能性は中程度</li> <li>• 9 - 正確である可能性が非常に低い一致</li> </ul>
ProcessedBy	<p>住所を処理した住所コーダーです。</p> <p><b>USA</b>                      米国住所コーダー</p> <p><b>CAN</b>                      カナダ住所コーダー</p> <p><b>INT</b>                      国際住所コーダー</p>
RecordType	<p>米国およびカナダの郵政当局が定義した住所レコードのタイプ (米国およびカナダの住所のみサポート):</p> <ul style="list-style-type: none"> <li>• FirmRecord</li> <li>• GeneralDelivery</li> <li>• HighRise</li> <li>• PostOfficeBox</li> <li>• RRHighwayContract</li> <li>• Normal</li> </ul>

応答要素	説明
RecordType.Default	"デフォルト" マッチを示すコード <b>Y</b> 住所はデフォルト レコードにマッチしています。 <b>NULL</b> 住所はデフォルト レコードにマッチしていません。
Status	マッチの成功または失敗。複数のマッチがある場合、一致する可能性のあるすべてのものに対してこのフィールドが "F" になります。 <b>NULL</b> 成功 <b>F</b> 失敗
Status.Code	失敗の原因 (ある場合)。複数のマッチがある場合、一致する可能性のあるすべてのものが "MultipleMatchesFound" になります。 <ul style="list-style-type: none"> <li>• DisabledCoder</li> <li>• InsufficientInputData</li> <li>• MultipleMatchesFound</li> <li>• UnableToValidate</li> </ul>
Status.Description	問題の説明 (ある場合)。 <b>Possible Multiple Addresses Found</b> Status.Code=MultipleMatchesFound の場合にこの値が表示されます。 <b>Address Not Found</b> Status.Code=UnableToValidate の場合にこの値が表示されます。 <b>PerformUSProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。 <b>PerformCanadianProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。 <b>PerformInternationalProcessing disabled</b> Status.Code=DisabledCoder の場合にこの値が表示されます。

## フィールドレベルの結果インジケータ

フィールドレベルの結果インジケータは、**ValidateAddress** が各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、

HouseNumber のフィールドレベルの結果インジケータは **HouseNumber.Result** に格納されます。

フィールドレベルの結果インジケータを有効にするには、OutputFieldLevelReturnCodes=Y を指定します。詳細については、[出力データ オプション](#) (573ページ) を参照してください。

次の表に、フィールドレベルの結果インジケータの一覧を示します。特定のフィールドが住所に適用されない場合、結果インジケータが空白になる場合があります。

**表 135 : フィールドレベルの結果インジケータ**

応答要素	説明
AddressRecord.Result	<p>これらの結果コードは国際住所のみに適用されます。</p> <ul style="list-style-type: none"> <li data-bbox="812 829 1421 934"><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。</li> <li data-bbox="812 955 1421 1018"><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</li> <li data-bbox="812 1039 1421 1071"><b>U</b> マッチしない。</li> <li data-bbox="812 1092 1421 1155"><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</li> </ul>

## 応答要素

## 説明

## ApartmentLabel.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国およびカナダの住所のみをサポートします。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
- R** アパートラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

## ApartmentNumber.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。カナダの住所のみ。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。EWS にマッチする米国の住所には、P の値が割り当てられます。米国およびカナダの住所のみをサポートします。
- R** アパート番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。



## 応答要素

## 説明

## City.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- F** ハイフンの欠落または句読文字エラー。カナダの住所のみ。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。
- R** 都市名が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## Country.Result

これらの結果コードは、米国またはカナダの住所には適用されません。

- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

## FirmName.Result

- C** 修正済み。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。米国およびカナダの住所のみをサポートします。
- U** マッチしない。米国およびカナダの住所のみをサポートします。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。米国住所にのみ適用されます。

## HouseNumber.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
- C** 修正済み。カナダの住所のみ。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- O** 範囲外。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。カナダの住所のみ。
- R** 家番号が必須ですが、入力住所から欠落しています。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

## LeadingDirectional.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。空白のない入力、空白のない値に修正されました。米国住所にのみ適用されます。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。カナダの住所には適用されません。

## 応答要素

## 説明

POBox.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
  - C** 修正済み。カナダの住所のみ。
  - D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
  - F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
  - M** 複数マッチ。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
  - P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。カナダの住所のみ。
  - R** 私書箱番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
  - S** 正規化。このオプションには、標準の略語が含まれます。
  - U** マッチしない。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
- 空白 不適用。

## 応答要素

## 説明

---

PostalCode.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
  - C** 修正済み。米国およびカナダの住所のみをサポートします。
  - F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
  - M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。カナダの住所には適用されません。
  - P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国住所には適用されません。
  - R** 郵便番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
  - S** 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。
  - U** マッチしない。例えば、通り名と郵便番号が一致しない場合、**StreetName.Result** と **PostalCode.Result** の両方に **U** が割り当てられます。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
-

## 応答要素

## 説明

## PostalCodeCity.Result

これらの結果コードは国際住所のみに適用されます。

- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## PostalCode.Source

これらの結果コードは米国住所のみに適用されます。

- FinanceNumber** 入力の ZIP Code™ は、USPS® Finance Number グループを使って検証されました。
- ZIPMOVE** 入力住所の ZIP Code™ は、USPS® が改訂した ZIP Code™ 境界に基づいて修正され、住所に別の ZIP Code™ が設定されました。

## PostalCode.Type

- P** ZIP Code™ には、PO Box 住所のみが含まれます。米国住所のみに適用されます。
- U** ZIP Code™ は、特定の会社または場所に割り当てられたユニークな ZIP Code™ です。米国住所のみに適用されます。
- M** ZIP Code™ は、軍施設の住所です。米国住所のみに適用されます。
- NULL** ZIP Code™ は、標準 ZIP Code™ です。

## 応答要素

## 説明

## RRHC.Result

- C** 修正済み。カナダの住所のみ。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- M** 複数マッチ。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。カナダの住所のみ。
- R** 地方配送路/幹線請負契約が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国およびカナダの住所のみをサポートします。
- U** マッチしない。米国およびカナダの住所のみをサポートします。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。米国およびカナダの住所のみをサポートします。

## RRHC.Type

これらの結果コードは米国住所にのみ適用されます。

- HC** 住所は、幹線請負契約住所です。
- RR** 住所は、地方配送路住所です。

## 応答要素

## 説明

## StateProvince.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国住所にのみ適用されます。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。
- R** アパート ラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## Street.Result

これらの結果コードは国際住所のみに適用されます。

- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
- P** パススルー。データは検証プロセスで使用されませんでしたでしたが、出力に保持されました。
- R** 通りが修正済みです。家番号が範囲外にあります。フランス、英国、および日本のレコードのみに適用。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。



## 応答要素

## 説明

---

StreetName.AbbreviatedAlias.Result

省略形エイリアス処理の結果を示します。次のいずれかです。

- NULL** 省略形エイリアス処理が実行されませんでした。
  - B** StreetName フィールドに基本通り名が格納されています。
  - L** 正規化された住所長が 31 文字未満なので、StreetName フィールドに基本名が格納されています。
  - N** 省略形エイリアスが見つかりませんでした。
  - Y** 省略形エイリアスが入力住所に見つかりました。StreetName フィールドに省略形エイリアスが格納されています。
-

## 応答要素

## 説明

---

StreetName.Alias.Type

この結果コードは米国住所にのみ適用されます。

注：以前のリリースでは、このフィールドは "Alias" と "Type" の間に "." がない StreetName.AliasType という名前でした。この古い名前は廃止されました。新しい名前 StreetName.Alias.Type を使用するよう、プロセスを更新してください。

- Abbreviated** エイリアスは通り名の省略形です。例えば、HARTS-NM RD は HARTSVILLE NEW MARLBORO RD の省略形エイリアスです。
- Changed** 通り名が正式に変更された場合に、新しい名前を表すエイリアスです。例えば、SHINGLE BROOK RD という通り名が CANNING DR に変更された場合、CANNING DR が変更済みのエイリアス タイプとなります。
- Other** この通り名エイリアスには、通りの他の名前や、通りの一般的な省略形などがあります。
- Preferred** 通り名エイリアスはその地域でよく使用されるエイリアスです。例えば、ある通りが "South Shore Dr." という名前なのは、湖の南岸を通っているためで、地方自治体の境界線の南にあるからではありません。この場合、"South" は前置方位記号ではないので、"S" と短縮してはいけません。したがって、"South Shore Dr." がよく使用されるエイリアスになります。
-

## 応答要素

## 説明

---

StreetName.PreferredAlias.Result

よく使用されるエイリアス処理の結果を示します。次のいずれかです。

- NULL** よく使用されるエイリアス処理が実行されませんでした。
  - A** 入力住所がエイリアスにマッチしたため、よく使用されるエイリアス処理が実行されませんでした。よく使用されるエイリアス処理は、基本住所に対してのみ実行されます。
  - N** よく使用されるエイリアスが見つかりませんでした。
  - Y** 入力住所に対してよく使用されるエイリアスが見つかりました。**StreetName** フィールドによく使用されるエイリアスが格納されています。
-

## 応答要素

## 説明

## StreetName.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国住所にのみ適用されます。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国住所には適用されません。
- S** 正規化。このオプションには、標準の略語が含まれます。米国およびカナダの住所のみをサポートします。
- U** マッチしない。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

## 応答要素

## 説明

---

StreetSuffix.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
  - C** 修正済み。米国およびカナダの住所のみをサポートします。
  - D** ドロップ済み入力に与えられたフィールドが削除されました。米国およびカナダの住所のみをサポートします。詳細については、「[その他の入力データについて](#)」を参照してください。
  - F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
  - M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
  - P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
  - S** 正規化。このオプションには、標準の略語が含まれます。
  - U** マッチしない。米国住所には適用されません。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
-

## 応答要素

## 説明

---

**TrailingDirectional.Result**

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- D** ドロップ済み入力に与えられたフィールドが削除されました。米国およびカナダの住所のみをサポートします。詳細については、「[その他の入力データについて](#)」を参照してください。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されません。
- P** パススルー。データは検証プロセスで使用されませんでしたが、出力に保持されました。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

---

**USUrbanName.Result**

これらの結果コードは米国住所にのみ適用されます。

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。
  - C** 修正済み。
  - M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。
  - U** マッチしない。
  - V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。
-

## オプションによる出力

ValidateAddress は、選択したオプションに応じて、追加のデータを返します。各オプションによって生成される出力については、以下のセクションのオプションの一覧を参照してください。

### Enhanced Line of Travel 出力

Enhanced Line of Travel 処理では、以下の出力を生成します。

応答要素	説明				
USLOTCode	Line of Travel の順序コードと、USPS® LOT 順序を表すインジケータ。このフィールドは、"nnnnY" という形式で、以下の要素で構成されます。 <table border="0" style="margin-left: 20px;"> <tr> <td><b>nnnn</b></td> <td>4 桁の LOT コード。</td> </tr> <tr> <td><b>Y</b></td> <td>次のいずれかです。               <ul style="list-style-type: none"> <li>• <b>A</b> — 昇順の LOT 順序</li> <li>• <b>D</b> — 降順の LOT 順序</li> </ul> </td> </tr> </table>	<b>nnnn</b>	4 桁の LOT コード。	<b>Y</b>	次のいずれかです。 <ul style="list-style-type: none"> <li>• <b>A</b> — 昇順の LOT 順序</li> <li>• <b>D</b> — 降順の LOT 順序</li> </ul>
<b>nnnn</b>	4 桁の LOT コード。				
<b>Y</b>	次のいずれかです。 <ul style="list-style-type: none"> <li>• <b>A</b> — 昇順の LOT 順序</li> <li>• <b>D</b> — 降順の LOT 順序</li> </ul>				
USLOTHex	昇順でのみファイルのソートが可能な 16 進値です。16 進値の範囲は、昇順で 0 ~ FF に続き、降順で FF ~ 0 です。				
USLOTSequence	アドオンの DPC の代わりに最終仕分けに使用される 2 バイトの値。大文字 1 文字の後に 0 ~ 9 の数字が 1 つ続きます。値の範囲は、A0 (99 降順) ~ J9 (00 降順)、および K0 (00 昇順) ~ T9 (99 昇順) です。				

LACS<sup>Link</sup> 出力

応答要素	説明
USLACS	<p>住所が、LACS<sup>Link</sup> 変換の候補であるかどうかを表します (米国住所のみ)。次のいずれかです。</p> <p><b>Y</b> 住所は LACS<sup>Link</sup> 処理の候補です。LACS<sup>Link</sup> が有効である場合は、ValidateAddress は、LACS<sup>Link</sup> データベースを使用して住所を変換します。変換に成功した場合、出力住所は LACS<sup>Link</sup> データベースから取得した新しい住所になります。変換できなかった場合は、住所は変換されません。</p> <p><b>N</b> 住所は LACS<sup>Link</sup> 処理の候補ではありません。ただし、LACS<sup>Link</sup> 処理が要求され、LACS<sup>Link</sup> データベースがインストールされており、かつ、次の条件のいずれかが満たされている場合は、LACS<sup>Link</sup> 処理が行われる場合があります。</p> <ul style="list-style-type: none"> <li>• 住所が地方配送路住所にマッチし、RecordType.Default フィールドで Y が返された場合。</li> <li>• 入力住所が、米国郵便データベースのいずれの住所にもマッチしなかった場合 (複数にマッチしたことによる失敗は、LACS<sup>Link</sup> の候補にはなりません)。</li> </ul>
USLACS.ReturnCode	<p>LACS<sup>Link</sup> 処理の成功または失敗を表します(米国住所のみ)。</p> <p><b>A</b> LACS<sup>Link</sup> 処理は成功しました。LACS<sup>Link</sup> 処理によってレコードがマッチしました。</p> <p><b>00</b> LACS<sup>Link</sup> 処理は失敗しました。LACS<sup>Link</sup> 処理において、マッチするレコードは見つかりませんでした。</p> <p><b>09</b> LACS<sup>Link</sup> 処理において、入力住所は、古い高層のデフォルト住所にマッチしました。住所は変換されています。不明確な住所の提供を避け、LACS<sup>Link</sup> 処理では、新しい住所を提供しません。</p> <p><b>14</b> LACS<sup>Link</sup> 処理は失敗しました。LACS<sup>Link</sup> 処理において、マッチする結果が検出されましたが、他の USPS<sup>®</sup> の規則に基づき、変換は行われませんでした。</p> <p><b>92</b> LACS<sup>Link</sup> 処理は成功しました。LACS<sup>Link</sup> 処理によってレコードがマッチしました。入力のユニット番号はドロップされました。</p> <p><b>NULL</b> LACS<sup>Link</sup> はレコードを処理しなかったか、または LACS<sup>Link</sup> 処理が実行されませんでした。</p>



## RDI 出力

応答要素	説明
RDI	住所の種類を表す値を返します。 <b>B</b> 住所は、企業住所です。 <b>R</b> 住所は、個人住所です。 <b>M</b> 住所は、個人住所であるとともに企業住所でもあります。 <b>NULL</b> 住所が ZIP + 4 <sup>®</sup> レベルでコード化されなかったか、または RDI™ が実行されなかったため、確認されていません。

## DPV および CMRA 出力

応答要素	説明
DPV	<p data-bbox="552 472 1136 504">Delivery Point Validation (DPV) 処理の結果を表します。</p> <p data-bbox="552 514 1234 546"><b>Y</b> DPV の確認済みです。この住所に郵便物を配達できます。</p> <p data-bbox="552 556 1023 588"><b>N</b> この住所に郵便物を配達できません。</p> <p data-bbox="552 598 1425 756"><b>S</b> 建物番号の妥当性は確認できましたが、ユニット番号は確認できませんでした。建物番号は、建物の主要な住所番号です。ユニット番号は、建物内のアパート、スイート、階など、各ユニットの郵便住所番号です。例えば、以下の住所の場合、<b>424</b> は建物番号、<b>12</b> はユニット番号です。</p> <p data-bbox="617 777 925 871">424 Washington Blvd.Apt.12 Oak Park IL 60302 USA</p> <p data-bbox="552 892 1425 1039"><b>D</b> 建物番号の妥当性は確認できましたが、ユニット番号は入力から欠落していました。建物番号は、建物の主要な住所番号です。ユニット番号は、建物内のアパート、スイート、階など、各ユニットの郵便住所番号です。例えば、以下の住所の場合、<b>424</b> は建物番号、<b>12</b> はユニット番号です。</p> <p data-bbox="617 1060 925 1155">424 Washington Blvd.Apt.12 Oak Park IL 60302 USA</p> <p data-bbox="552 1176 1169 1207"><b>M</b> 住所は複数の有効な配達ポイントにマッチします。</p> <p data-bbox="552 1228 1425 1291"><b>U</b> 住所は、ZIP + 4<sup>®</sup> レベルでコード化されなかったため、確認できませんでした。</p> <p data-bbox="552 1312 1023 1344"><b>V</b> 住所は、誤検出違反を起こしました。</p>
CMRA	<p data-bbox="552 1438 1425 1501">住所が Commercial Mail Receiving Agency (CMRA: 民間私書箱) であることを表します。</p> <p data-bbox="552 1512 925 1543"><b>Y</b> 住所は CMRA です。</p> <p data-bbox="552 1564 1039 1596"><b>N</b> 住所は CMRA ではありません。</p> <p data-bbox="552 1617 893 1648"><b>U</b> 確認できません。</p>

## 応答要素

## 説明

## DPVFootnote

DPV 補足コード。

- AA** 入力住所は、ZIP + 4<sup>®</sup> ファイルにマッチしました。
- A1** 入力住所は、ZIP + 4<sup>®</sup> ファイルにマッチしませんでした。
- BB** 入力住所は、DPV にマッチしました (すべてのコンポーネント)。
- CC** 入力住所の主要な番号は DPV にマッチしましたが、補助的な番号はマッチしませんでした (存在しましたが有効ではありませんでした)。
- F1** 入力住所は軍関係の住所。DPV は省かれます。
- G1** 入力住所は一般的な配達住所。DPV は省かれます。
- M1** 入力住所の主要な番号が欠落しています。
- M3** 入力住所の主要な番号が無効です。
- N1** 入力住所の主要な番号は DPV にマッチしましたが、高層住所に補助的な番号が欠落しています。
- P1** 入力住所に、RR または HC Box 番号がありません。
- P3** 入力住所に、PO、RR、または HC Box 番号がありません。
- RR** 入力住所は、CMRA にマッチしました。
- R1** 入力住所は CMRA にマッチしましたが、補助的な番号が存在しません。
- R7** 入力住所は擬似キャリアルート R777 にマッチしました (通り配達の対象になりません)。
- U1** 入力住所はユニーク ZIP。DPV は省かれます。

## DPVVacant

建物が空家 (90 日間使用されていない)かどうかを表します。次のいずれかです。

- Y** 建物は空家です。
- N** 建物は空家ではありません。
- NULL** [DPVDetermineVacancy] オプションが選択されていません。

## 応答要素

## 説明

DPVNoStat

建物が、郵便物を受け取ることのできない "no stat" の建物であるかどうかを表します。次のいずれかです。

- Y** 建物は、郵便物を受け取ることのできない "no stat" の建物です。
- N** 建物は、郵便物を受け取ることのできない "no stat" の建物ではありません。
- NULL** DPVDetermineNoStat オプションが選択されていません。

Suite<sup>Link</sup> 出力

## 応答要素

## 説明

SuiteLinkReturnCode

ValidateAddress が、補助的な住所情報を修正したかどうかを表します (米国住所のみ)。次のいずれかです。

- A** ValidateAddress は、補助的な住所情報を修正しました。
- 00** ValidateAddress は、補助的な住所情報を修正しませんでした。
- NULL** Suite<sup>Link</sup> は実行されませんでした。
- XX** Suite<sup>Link</sup> 処理においてエラーが発生しました。例えば、Suite<sup>Link</sup> データベースの有効期限が切れている場合にエラーが発生します。

## 応答要素

## 説明

## SuiteLinkMatchCode

Suite<sup>Link</sup> のマッチング処理に関する追加情報を提供します(米国住所のみ)

- A** ValidateAddress は、補助的な住所情報を修正しました。
  - B** ValidateAddress は、補助的な住所情報を修正しませんでした。マッチング処理に関するその他の詳細情報はありません。
  - C** FirmName フィールドの語はすべて、「ノイズ」語です。ノイズ語は、USPS<sup>®</sup> によって定義されており、企業名のマッチングの際には無視されます。ノイズ語の例としては、"company" や "corporation" があります。ValidateAddress は、ノイズ語のみで構成される企業名に対し、補助的な住所情報を修正できません。例えば、"Company and Corporation" という表記は、ノイズ語のみで構成されています。
  - D** 住所は、高層のデフォルト住所ではありません。Suite<sup>Link</sup> マッチングは、高層のデフォルト住所に対してのみ行われます。高層デフォルトとは、住所に有効な補助的な情報が含まれていない(アパート番号やアパート種別が欠落している)場合に使用されるデフォルトです。
  - E** Suite<sup>Link</sup> データベースの有効期限が切れているため、Suite<sup>Link</sup> 処理は失敗しました。
- NULL** Suite<sup>Link</sup> は実行されなかったか、エラーが発生しました。

## SuiteLinkFidelity

ValidateAddress における、Suite<sup>Link</sup> データベースの企業名に対するマッチング精度を表します。

- 1** 企業名は、Suite<sup>Link</sup> データベースに完全に一致しました。
  - 2** 精度の高いマッチです。企業名に含まれる語が、1 語を除いてすべて Suite<sup>Link</sup> データベースの企業名に一致しました。
  - 3** 精度の低いマッチです。企業名の中の複数の語が、Suite<sup>Link</sup> データベースの企業名に一致しませんでした。
- NULL** Suite<sup>Link</sup> が企業名のマッチングに失敗したか、実行されなかったか、またはエラーが発生しました。

## VeriMove 出力

応答要素	説明
VeriMoveDataBlock	<p>ValidateAddress が、VeriMove Express に渡される入力データを含む 250 バイトのフィールドを返すかどうかを示します。このフィールドには、VeriMove で必要とされる詳細結果インジケータ データが含まれます。このフィールドの内容については、VeriMove のユーザー ガイドを参照してください。次のいずれかです。</p> <p><b>Y</b>          フィールド VeriMoveDataBlock を返します。</p> <p><b>N</b>          フィールド VeriMoveDataBlock を返しません。</p>

**AdditionalInputData** について

ValidateAddress は、住所正規化プロセスにおいて、一部の入力データを無視します。この余分なデータ ("ドロップ データ" と呼ばれることもあります) は、AdditionalInputData に返されます。ドロップ データの例としては、次のものがあります。

- 配達指示 ("勝手口に置いてください" など)
- 電話番号 ("555-135-8792" など)
- 注意書き ("Attn: John Smith" など)

このようなデータは通常、住所に混在していることはありません。混在している場合、ValidateAddress はほとんどの場合にこの余分なデータを認識することができ、AdditionalInputData に返します。

**注:** ValidateAddress は、split indicia 住所からのドロップ データを返しません。split indicia 住所とは、主要な住所が複数の住所行に分割されている住所のことです。例えば、主要な住所が "1 Green River Valley Rd" である場合、次のようになります。

```
1 Green River
Valley Rd
01230
```

住所に複数のドロップ データがある場合、各データは、米国住所の場合はセミコロンと空白 ("; ")、米国外の住所の場合は空白で区切られます。AdditionalInputData におけるドロップ データの順序は、次のようになります。

1. care of (気付)、mail stop (郵便物集配所)(米国住所のみ)
2. 住所行に検出されたその他の余分なデータ
3. まったく未使用のデータ行

例えば、入力住所が次のとおりであるとします。

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

この場合、AdditionalInputData には次のデータが含まれます。

C/O John Smith; Apt 5 Drop At Back Dock; 555-123-4567; Jsmith@g1.Com; 555-123-4567

ValidateAddress では、以下の種類の余分なデータを処理できます。

#### Care Of (気付) データ

米国住所に対しては、"care of" データが AdditionalInputData に返されます。以下の住所には、"care of" データの例が含まれています。

123 Main St C/O John SmithApt 505674

123 Main StApt 5 ATTN John Smith05674

123 Main St Apt 5MailStop 205674

#### 独立した住所行に存在する余分なデータ

ValidateAddress は、米国およびカナダの住所に対し、独立した住所行に余分なデータを返します。

米国住所に対しては、住所行の空白でない最初の 2 行を使用して、住所の正規化を行います。ただし、企業名抽出または都市化コード抽出のオプションが有効である場合を除きます (詳細については、「[米国住所の住所行処理 \(572ページ\)](#)」を参照してください)。他の住所行に存在するデータは、AdditionalInputData に返されます。以下の住所において、"John Smith" は AdditionalInputData に返されます。"John Smith" は空白でない 3 つめの住所行に存在しており、ValidateAddress は米国住所に対して、空白でない最初の 2 つの住所行のみを使用するためです。

123 Main StApt 5John Smith05674

空白でない最初の 2 つの住所行に余分なデータが含まれる場合、そのデータは AdditionalInputData に返されます。例えば、以下の住所において、"John Smith" は AdditionalAddressData に返されます。

123 Main StJohn Smith05674

John Smith123 Main St05674

以下の住所では、"John Smith" と "Apt 5" の両方が AdditionalAddressData に返されます。"John Smith" が返されるのは、これが、最初の 2 つの住所行の 1 つに存在する余分なデータであるためです。"Apt 5" が返されるのは、米国住所データは、空白でない最初の 2 行に記載しなければならないためです。

John Smith123 Main StApt 505674

## 住所行に混在する余分なデータ

住所行に混在する余分なデータは、`AdditionalInputData` に返されます。例えば、以下の住所において、`"John Smith"` は `AdditionalInputData` に返されます。

123 Main St John Smith05674

123 Main St Apt 5 John Smith05674

123 Main St John SmithApt 505674

123 Main StApt 5 John Smith05674

米国住所に対しては、住所行の末尾に存在する余分なデータのみが `AdditionalInputData` に返されます。米国住所において、住所行の末尾以外に存在する余分なデータは返されません。例えば、以下の住所において、`"John Smith"` は返されません。

John Smith 123 Main St05674

123 Main John Smith St05674

マッチさせるために通り名が変更され、通り名または接尾語が行の末尾にあった場合は、`AdditionalInputData` には、元の通り名や接尾語が含まれることがあります。例えば、次の住所があるとします。

Pitney Bowes  
4200 ParliamentLanham MD

`ValidateAddress` は、ストリート名の綴りを修正し、接尾語を追加して、修正済みのストリーットの住所として `"4200 Parliament Pl"` を返し、`"Parliament"` を `AdditionalInputData` に返します。

## 二重住所

二重住所とは、通り情報と、PO Box/地方配送路/幹線請負契約情報の両方を含む住所のことです。選択した処理オプションに応じて、住所の正規化に使用されない二重住所の一部が、`AdditionalInputData` に返される場合があります。詳細については、[二重住所ロジックについて](#) (588ページ) を参照してください。

## `ValidateAddressGlobal`

`ValidateAddressGlobal` は、米国およびカナダ以外の住所の標準化と検証のみを制御します。`ValidateAddressGlobal` は、米国およびカナダの住所の妥当性も確認できますが、その他の国の住所の妥当性を確認する能力に優れています。米国およびカナダ以外の住所を大量に処理する場合は、`ValidateAddressGlobal` の使用を検討してください。

`ValidateAddressGlobal` は `Universal Addressing` モジュールの一部です。

`ValidateAddressGlobal` は、書き直し、パーシング、バリデーション、書式設定など、いくつもの手順を実行して、住所の品質を高めています。



### 文字セットのマッピングと書き直し

`ValidateAddressGlobal` は他国の文字列と、それらの複雑な問題を処理します。Unicode に完全対応の文字列処理を使用するため、アルファベット以外の文字をラテン文字セットに書き直したり、異なる文字セット間でマッピングしたりできます。

文字セットのマッピング、および書き直しについて、以下の機能があります。

- UTF-8、ISO 8859-1、GBK、BIG5、JIS、EBCDIC など、30 以上の文字セットをサポート
- 言語のルールに従って、付加記号を正しく "除去"
- さまざまなアルファベットをラテン スクリプトに書き直し
- ギリシャ文字 (BGN/PCGN 1962、ISO 843 - 1997)
- キリル文字 (BGN/PCGN 1947、ISO 9 - 1995)
- ヘブライ語
- 日本語の片仮名、平仮名、漢字
- 中国語のピンイン (標準中国語、 広東語)
- 韓国語のハングル文字

### 住所のパーシング、書式設定、および正規化

住所データのフィールド入力の誤りを再構成することは、特に他国の住所で行う場合、複雑で難しい作業です。住所データをコンピュータのシステムに入力する際、曖昧になってしまう部分が多いからです。特に問題なのが、(企業や個人名を通りの住所フィールドに入力するなど) 要素を誤ったフィールドに入力したり、省略形を使用する場合に、言語固有だけでなく、国固有の省略形に変えてしまうケースです。`ValidateAddressGlobal` は住所行の住所要素を識別し、正しいフィールドに割り当てます。これは実際の検証前に行う重要な作業です。再構成を行わなければ、"一致が見つからない" という結果になる可能性があります。

住所要素の正しい識別は、特定のフィールド長要件に合わせて住所を切り捨てたり、短縮しなければならない場合にも重要です。正しい情報が正しいフィールドに割り当てられていれば、特定の切り捨てルールを適用することができます。

- 住所行をパースおよび解析し、個々の住所要素を識別
- 30 を越える文字セットを処理
- 宛先国の郵便ルールに従って住所の書式を整える
- 住所要素を正規化 (AVENUE を AVE に変更するなど)

### Global Address 検証

住所の検証は、正しくパースされた住所データを郵便組織または他のデータ プロバイダが提供する参照データベースと比較する訂正処理です。`ValidateAddressGlobal` は、洗練されたファジー マッチング テクノロジーを使用して個々の住所要素を検証し、正しいことを確認するとともに、郵便規格とユーザの優先設定に基づいて出力を正規化および書式設定します。`FastCompletion` 検

証タイプは、簡易住所入力アプリケーションに使用できます。いくつかの住所フィールドには切り捨てられたデータを入力することができ、この入力に基づいて提案を生成します。

住所を完全に検証できない場合もあります。**ValidateAddressGlobal** には、配達可能性によって住所を分類する、ユニークな配達可能性評価機能があります。

### リソース URL

```
http://server:port/soap/ValidateAddressGlobal
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:val="http://www.pb.com/spectrum/services/ValidateAddressGlobal">
  <soapenv:Header/>
  <soapenv:Body>
    <val:ValidateAddressGlobalRequest>
      <val:input>
        <val:Address>
          <val:Country>USA</val:Country>
          <val:AddressLine1>1 Global View</val:AddressLine1>
          <val:City>Troy</val:City>
          <val:StateProvince>NY</val:StateProvince>
          <val:PostalCode></val:PostalCode>
        </val:Address>
      </val:input>
    </val:ValidateAddressGlobalRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:ValidateAddressGlobalResponse
xmlns:ns2="http://spectrum.pb.com/"

xmlns:ns3="http://www.pb.com/spectrum/services/ValidateAddressGlobal">
      <ns3:output>
        <ns3:Address>
          <ns3:Country>UNITED STATES</ns3:Country>
          <ns3:AddressLine1>1 GLOBAL VW</ns3:AddressLine1>
          <ns3:HouseNumber>1</ns3:HouseNumber>
          <ns3:StreetName>GLOBAL</ns3:StreetName>
          <ns3:StreetSuffix>VW</ns3:StreetSuffix>
          <ns3:City>TROY</ns3:City>
          <ns3:PostalCode>12180-8371</ns3:PostalCode>
```

```

<ns3:PostalCode.Base>12180</ns3:PostalCode.Base>
<ns3:PostalCode.AddOn>8371</ns3:PostalCode.AddOn>
<ns3:StateProvince>NY</ns3:StateProvince>
<ns3:County>RENSSELAER</ns3:County>
<ns3>LastLine>TROY NY 12180-8371</ns3>LastLine>
<ns3:AddressBlock1>1 GLOBAL VW</ns3:AddressBlock1>
<ns3:AddressBlock2>TROY NY 12180-8371</ns3:AddressBlock2>

<ns3:ProcessStatus>C4</ns3:ProcessStatus>
<ns3:ProcessStatus.Description>
  Corrected - all elements have been checked
</ns3:ProcessStatus.Description>
<ns3:ModeUsed>BATCH</ns3:ModeUsed>
<ns3:CountOverflow>NO</ns3:CountOverflow>
<ns3:MailabilityScore>5</ns3:MailabilityScore>
<ns3:Confidence>85.09</ns3:Confidence>

<ns3:ElementResultStatus>88F0F8E0F000000000E0</ns3:ElementResultStatus>

<ns3:ElementInputStatus>00606050600000000060</ns3:ElementInputStatus>

<ns3:ElementRelevance>11101010100000000010</ns3:ElementRelevance>
  <ns3:AddressType>S</ns3:AddressType>
  <ns3:AMAS.Status>EAM0</ns3:AMAS.Status>
  <ns3:user_fields/>
</ns3:Address>
</ns3:output>
</ns3:ValidateAddressGlobalResponse>
</soap:Body>
</soap:Envelope>

```

## 要求

### 入力データのパラメータ

`ValidateAddressGlobal` は、入力として標準住所を受け取ります。どの国の住所であるかにかかわらず、すべての住所がこのフォーマットを使用します。

表 136 : ValidateAddressGlobal の入力

パラメータ	書式	説明
AddressLine1 から AddressLine6	文字列 [79]	これらのフィールドには住所行データが格納されます。AddressLine1 には最初の住所行、AddressLine2 には 2 行目の住所行が格納されます。以降もこれと同様になります。都市、州/省、および郵便番号情報は、住所行フィールドではなく、それぞれのフィールドに配置する必要があります。例:  <b>AddressLine1:</b> 17413 Blodgett Road <b>AddressLine2:</b> PO Box 123 <b>City:</b> Mount Vernon <b>StateProvince:</b> WA <b>PostalCode:</b> 97273 <b>Country:</b> USA  入力住所が適切な住所行および City、StateProvince、PostalCode フィールドにまだパースされていない場合は、住所行フィールドの代わりに UnformattedLine フィールドを使用してください。
City	文字列 [79]	都市名
StateProvince	文字列 [79]	州または省。
PostalCode	文字列 [79]: 99999 99999-9999 A9A9A9 A9A 9A9 9999 999	住所の郵便番号。米国では、ZIP Code®になります。
Contact	文字列 [79]	受取人の名前。例えば、"Mr.Jones"。
Country	文字列 [79]	国名。または オプションに値が指定されていない場合、国を指定する必要があります。
FirmName	文字列 [79]	会社名または企業名。

パラメータ	書式	説明
Street	文字列 [79]	Street
Number	建物 [79]	番号
Building	文字列 [79]	建物
SubBuilding	文字列 [79]	SubBuilding
DeliveryService	文字列 [79]	配送サービス
UnformattedLine1 から UnformattedLine10	文字列 [79]	<p>入力住所が完全にパースされていない場合、かつ <code>ValidateAddressGlobal</code> によって住所を適切なフィールドにパースしたい場合は、このフィールドを使用してください。例:</p> <p><b>UnformattedLine1:</b> 17413 Blodgett Road<b>UnformattedLine2:</b> PO Box 123<b>UnformattedLine3:</b> Mount Vernon WA 97273<b>UnformattedLine4:</b> USA</p> <p>この住所は、以下の出力フィールドにパーシングされます。</p> <p><b>AddressLine1:</b> 17413 Blodgett Road<b>AddressLine2:</b> PO Box 123<b>City:</b> Mount Vernon<b>StateProvince:</b> WA<b>PostalCode:</b> 97273<b>Country:</b> USA</p> <p>注: フォーマットされていない行フィールドへの入力を指定した場合は、住所全体をフォーマットされていない行フィールドだけを使用して指定する必要があります。City や StateProvince など、その他のフィールドを、フォーマットされていない行フィールドと合わせて使用することはできません。</p>

## オプションのパラメータ 入力オプション

表 137 : ValidateAddressGlobal の入力オプション

パラメータ	説明/有効値
Database.AddressGlobal	住所検証で使用する郵便データを含むデータベースリソースを指定します。指定できるのは、Management Console の【グローバル データベース リソース】パネルで定義されたデータベースに限られます。詳細については、『Spectrum™ Technology Platform 管理ガイド』を参照してください。
Input.DefaultCountryISO3	入力レコードに明示的な国情報が含まれない場合に使用するデフォルトの国を指定します。指定する際、ISO3 の国コードを使用してください。デフォルトの国を指定しない場合、各入力レコードの Country 入力フィールドに国を指定する必要があります。ISO コードの一覧は、「ISO 国コードとモジュール サポート (735ページ)」を参照してください。
Input.ForceCountryISO3	住所レコードが、常にここで指定された国から発送されたものとして扱われるようになります。住所レコードの国やデフォルトの国は上書きされます。指定する際、ISO3 の国コードを使用してください。ISO コードの一覧は、ISO 国コードとモジュール サポート (735ページ) を参照してください。
Input.FormatDelimiter	<p>入力ファイルで複数行の住所に標準以外の書式設定を使用できます。このフィールドで指定できる値を次に示します。</p> <ul style="list-style-type: none"> <li>• CRLF (デフォルト)</li> <li>• LF</li> <li>• CR</li> <li>• SEMICOLON ( 2101 MASSACHUSETTS AVE NW ; WASHINGTON DC 20008)</li> <li>• COMMA (2101 MASSACHUSETTS AVE NW , WASHINGTON DC 20008 )</li> <li>• TAB (2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008 )</li> <li>• PIPE (2101 MASSACHUSETTS AVE NW   WASHINGTON DC 20008 )</li> <li>• SPACE (2101 MASSACHUSETTS AVE NW WASHINGTON DC 20008)</li> </ul> <p>注：入力オプションと出力オプションの両方に同じ値を選択する必要があります。</p>

## 出力オプション

表 138 : ValidateAddressGlobal の出力オプション

パラメータ	説明
Result.MaximumResults	このオプションには、返される候補住所の最大数を指定します。このフィールドは、バッチ処理には無効です。それ以外の処理モードの場合、デフォルト値は1、最大値は99です。FastCompletionモードを使用している場合は、1よりも大きい数を入力して、フィールド入力のための選択肢が複数返されるようにするとよいでしょう。
Result.IncludeInputs	出力の中に、入力データを含めるかどうかを指定します。有効にすると、.Inputで終わるフィールド(対応する入力フィールドを含む)が出力に含まれます。例えば、出力フィールドの AddressLine1.Input に、入力フィールドの AddressLine1 に指定したデータが含まれます。 <b>TRUE</b> 入力データを出力に含めます。 <b>FALSE</b> 出力に元の入力データを含めません (デフォルト)。
Result.StateProvinceType	StateProvince フィールドのフォーマットを指定します。次のいずれかを選択します。 <b>ABBREVIATION</b> 州または省の省略形を返します。例えば、North Carolina であれば "NC" が返ります。 <b>COUNTRY_STANDARD</b> その国の郵便当局で使用されるフォーマットに応じて、省略形または完全な名前が返ります (デフォルト)。 <b>EXTENDED</b> 州または省の (省略形ではなく) 完全な名前が返ります (例えば、"North Carolina" など)。

## パラメータ

## 説明

---

Result.CountryType	ValidateAddressGlobal から返される国名で使用する言語またはコードを指定します。
<b>ISO2</b>	その国の 2 文字の ISO コード
<b>ISO3</b>	その国の 3 文字の ISO コード
<b>ISO_NUMBER</b>	ISO 国番号
<b>NAME_CN</b>	中国語
<b>NAME_DA</b>	デンマーク語
<b>NAME_DE</b>	ドイツ語
<b>NAME_EN</b>	英語 (デフォルト)
<b>NAME_ES</b>	スペイン語
<b>NAME_FI</b>	フィンランド語
<b>NAME_FR</b>	フランス語
<b>NAME_GR</b>	ギリシャ文字
<b>NAME_HU</b>	ハンガリー語
<b>NAME_IT</b>	イタリア語
<b>NAME_JP</b>	日本語
<b>NAME_KR</b>	韓国語
<b>NAME_NL</b>	オランダ語
<b>NAME_PL</b>	ポーランド語
<b>NAME_PT</b>	ポルトガル語
<b>NAME_RU</b>	ロシア語
<b>NAME_SA</b>	サンスクリット語
<b>NAME_SE</b>	スウェーデン語

---



## パラメータ

## 説明

**Result.PreferredScript**

出力がどのアルファベットで返されるかを指定します。データがどのアルファベットで返されるかは、国によって異なります。言語の設定で何を選択したかに関わらず、ほとんどの国で出力は **Latin I** になります。

<b>ASCII_Extended</b>	特殊文字 (Ã– = OE など) で拡張された ASCII 文字
<b>ASCII_Simplified</b>	ASCII 文字
<b>Database</b>	(デフォルト) Latin I または ASCII 文字 (参照データベースの標準による)
<b>Latin</b>	Latin I 文字
<b>Latin_Alt</b>	Latin I 文字 (代替の書き直し)
<b>Postal_Admin_Alt</b>	Latin I または ASCII 文字 (現地郵便局の代替)
<b>Postal_Admin_Pref</b>	Latin I または ASCII 文字 (現地郵便局の選択による)

Latin I 以外のアルファベットを使用する国の場合、返されるアルファベットは国によって異なります。詳細については、[Latin 1 以外のアルファベットを使用する国](#) (666ページ) を参照してください。

**Result.PreferredLanguage**

出力がどの言語で返されるかを指定します。データがどのアルファベットで返されるかは、国によって異なりますが、ほとんどの国で、言語の設定に何を選択したかに関わらず、出力は **Latin I** になります。

<b>DATABASE</b>	各住所の参照データから得られた言語で返されます。こちらがデフォルトです。
<b>ENGLISH</b>	地方および州/省の名前が (可能であれば) 英語で出力されます。

**Result.Casing**

出力の大文字と小文字の区別を指定します。

<b>NATIVE</b>	参照データベースの標準に基づいて出力されます。
<b>UPPER</b>	すべての国で、大文字で出力されます。
<b>LOWER</b>	すべての国で、小文字で出力されます。
<b>MIXED</b>	国固有のルールに従って大文字と小文字が判断されます。
<b>NOCHANGE</b>	パースモードの場合、データは入力された方法で返されます。バリデーションモードの場合、大文字と小文字の区別には参照データに見つかったものや、郵便のルールに従ったものを使用します。参照データを使用してチェックできなかった値は、入力時の大文字と小文字の区別を保持します。

## パラメータ 説明

**Result.FormatDelimiter** 出力で複数行の住所に標準以外の書式設定を使用できます。このフィールドで指定できる値を次に示します。

- CRLF (デフォルト)
- 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)

注：入力オプションと出力オプションの両方に同じ値を選択する必要があります。

### Latin 1 以外のアルファベットを使用する国

Latin I 以外のアルファベットを使用する国の場合、返されるアルファベットは国によって異なります。それらの国々で、出力がどのように返されるかを以下の表に示します。ここに含まれないすべての国では、フィールド オプションに指定された値が使用されます。

Country	データ	ベース	ラテン文字
RUS	キリル文字	キリル文字	キリル文字
			CYRILLIC_ISO CYRILLIC_BGN CYRILLIC_ISO + LATIN CYRILLIC_ISO + LATIN_SIMPLE
JPN	漢字	漢字	かな
			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

Country データ  
ベース

ラテン文字

TWN	Hanzi	Hanzi	Hanzi	CHINESE_ CANTONESE	CHINESE_ MANDARIN	CHINESE_ CANTONESE + LATIN_SIMPLE	CHINESE_ CANTONESE + LATIN
GRC	ギリシャ文 字	ギリシャ文 字	ギリシャ文 字	GREEK_ISO	GREEK_BGN	GREEK_ISO + LATIN_SIMPLE	GREEK_ISO + LATIN
KOR	ラテン文字	ハングル	Hanja	KOREAN	KOREAN	KOREAN + LATIN_SIMPLE	KOREAN + LATIN
ISR	ラテン文字	ヘブライ語	ヘブライ語	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

プロセス オプション

表 139 : ValidateAddressGlobal のプロセス オプション

パラメータ	説明
Process.OptimizationLevel	<p data-bbox="495 441 1427 514">このオプションを使用して、処理速度と品質が適切なバランスを保つよう設定します。次のいずれかです。</p> <p data-bbox="495 525 1427 598"><b>NARROW</b> パーサーは、House Number を Street の情報から分割することを除き、入力の割り当てに厳密に従います。</p> <p data-bbox="495 609 1427 934"><b>STANDARD</b> パーサーは、以下に示すように住所要素をよりアクティブに分割します。</p> <ul data-bbox="633 661 1169 934" style="list-style-type: none"> <li>• Province は Locality の情報から分割される</li> <li>• PostalCode は Locality の情報から分割される</li> <li>• House Number は Street の情報から分割される</li> <li>• SubBuilding は Street の情報から分割される</li> <li>• DeliveryService は Street の情報から分割される</li> <li>• SubBuilding は Building の情報から分割される</li> <li>• Locality は PostalCode の情報から分割される</li> </ul> <p data-bbox="495 945 1427 1060"><b>WIDE</b> パーサーによる分割は Standard に似た方法で行われますが、追加で最大 10 のパーシング候補が検証に渡されて処理されます。検証は検索のツリーを広げ、参照データのエントリを対象に追加してマッチングを行います。</p> <p data-bbox="495 1071 1427 1144">最適化レベルを調整しても、ここで説明したような分割で必要とされる郵便参照データ情報が揃っていない国においては、効果が得られない場合があります。</p> <p data-bbox="495 1155 1427 1291">分類の精度を Narrow から Standard に増やしても処理パワーはある程度消費されますが、最適化レベルを Wide に設定し、検索ツリーを大きくして検証を行った場合、指定された入力データから最も精密な結果を得ようとデータのアクセスや比較の件数が増加するため、処理速度が非常に遅くなります。</p>

## パラメータ 説明

---

Process.Mode	住所に対して実行する処理のタイプを指定します。次のいずれかです。
<b>BATCH</b>	<p>このモードは手動でのデータの入力または選択が不可能なバッチ処理環境で使用します。処理速度を重視して最適化されているため、自動で修正できないあいまいなデータが見つかった場合は、住所修正の試行が停止します。指定の国がデータベースに見つからない場合、バッチ処理モードはパースモードに変更されます。</p> <p>注：プロセスステータスとして I3 の値が返された場合、試行は失敗とみなされ、ステータスとして F の値が返されます。</p>
<b>CERTIFIED</b>	<p>このモードはオーストラリア郵便当局のバッチ処理環境で使用します。Validate Address Global はオーストラリア郵便公社の Address Matching Approval System (AMAS) で認定されています。これは郵便住所ファイルに照らして郵便の正規化と検証を行うことで、郵便料金の割引と不達郵便物の最小化を実現します。</p>
<b>FASTCOMPLETION</b>	<p>高速実行モードは、切り捨て処理されたデータを住所フィールドに入力して、Validate Address Global による提案の生成を行いたい場合に使用します。例えば、コールセンターや店頭などで作業をしている場合に住所要素の一部だけを入力し、高速実行の機能を使用して、完全な住所の候補を正しく入手することができます。</p>
<b>INTERACTIVE</b>	<p>このモードは、インタラクティブな環境で処理を行い、住所入力があいまいな場合に提案を生成する場合に使用します。このタイプの検証は、とりわけ顧客や潜在顧客から入手したデータを入力した環境で使用されます。住所がほぼ完全に入力されている必要があり、その入力データの検証や修正を試行します。あいまいなデータが検出された場合、この検証タイプでは最大で 20 の提案が生成され、それを選択候補の一覧として使用できます。指定の国がデータベースに見つからない場合、インタラクティブモードはパースモードに変更されます。</p>
<b>PARSE</b>	<p>このモードは住所入力をトークンに分割し、そのあとに他のシステムで行われる処理に送る場合、すなわち検証をバイパスする場合に使用します。例えば、住所データの品質がすでに高く、単に素早くトークンに分割して外部のシステムにエクスポートするか、あるいは下流のステージで使用する場合にこのモードを利用できます。</p>

---



応答要素	説明
AddressLine1-6	<p>住所が検証された場合、住所行フィールドには検証済みで正規化済みの住所行が入ります。住所が検証できなかった場合、住所行フィールドには入力された住所が変更されずに入ります。ただし、住所の最後の行は LastLine フィールドに入ります。例:</p> <p>AddressLine1: 4200 PARLIAMENT PL STE 600LastLine: LANHAM MD 20706-1882</p>
AdministrativeDistrict	州/省より小さいが、都市よりも大きいエリア
ApartmentLabel	フラットまたはユニット タイプ (STE や APT など)。例: 123 E Main St <b>Apt 3</b>
ApartmentNumber	フラットまたはユニット番号。例: 123 E Main St <b>Apt 3</b>
BlockName	地所やブロックの名前。
BuildingName	建物の名前。例えば <b>Sears Tower</b> など。
City	都市の名前。例えば <b>Vancouver, BC</b> など。
City.AddInfo	都市の追加情報。
City.SortingCode	例えばプラハやダブリンなど、特定の国の広い地方で、郵便当局が配達時間を短縮するために使用するコード。
Contact	受取人の名前。例えば、 <b>Mr. Jones</b> など。
Country	国が、言語または オプションで指定されたコードで入ります。
County	州や省に従属する情報。州や省をさらに細かく分割します。米国の郡はその一例です。
FirmName	会社名。
Floor	部屋番号やアパート番号など、建物をさらに細かく分割する情報。例: 123 E Main St Apt 3, <b>4th Floor</b>

応答要素	説明
HouseNumber	家番号 1。例: 298A-1B New South Head Rd
LastLine	完成された最終の住所行 (都市、州/省、および郵便番号)。
LeadingDirectional	通り名の前に付けて通りの方向を表します。例えば、138 N Main Street の N がこれに該当します。
Locality	場所の名前に従属し、地方をさらに細かく分割します。例としては、メキシコの Colonia や、スペインの Urbanisaciones などがあります。
POBox	郵便受けの記述子 (POBox、Postfach、Case Postale など) と番号。
PostalCode	住所の郵便番号。郵便番号のフォーマットは国によって異なります。
PostalCode.AddOn	郵便番号の 2 番目の部分。例えば、カナダの住所ではこれは LDU です。米国住所に対しては、これは ZIP+4 アドオンです。ほとんどの国ではこのフィールドを使用しません。
PostalCode.Base	郵便番号の基本部分。
Room	建物の部屋番号。
SecondaryStreet	補助的な通りまたは地方集配路の名前。
StateProvince	州または省の名前。
StreetName	建物が存在する通りの名前。例: 123 E <b>Main</b> St Apt 3
StreetSuffix	通り接尾語。例: 123 E Main <b>St</b> Apt 3
SubBuilding	部屋番号など建物の一部。例えば Suite 102 など。
Suburb	場所の名前に従属し、地方をさらに細かく分割します。例えばトルコの Mahalle など。



応答要素	説明
Territory	地域の名前。地域は州/省より大きいものです。
TrailingDirectional	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>

### 元の入力データ

このオプションは、元の入力データを <フィールド名>.Input フィールドに出力します。

**表 141 : 元の入力データ**

応答要素	書式	説明
AddressLine1.Input	文字列 [79]	最初の住所行。
AddressLine2.Input	文字列 [79]	2 行目の住所行。
AddressLine3.Input	文字列 [79]	3 行目の住所行。
AddressLine4.Input	文字列 [79]	4 行目の住所行。
AddressLine5.Input	文字列 [79]	5 行目の住所行。
AddressLine6.Input	文字列 [79]	6 行目の住所行。
City.Input	文字列 [79]	都市名

応答要素	書式	説明
StateProvince.Input	文字列 [79]	州または省
PostalCode.Input	文字列 [79]:	住所の郵便番号。米国では、ZIP Code になります。次のいずれかのフォーマットです。  99999 99999-9999 A9A9A9 A9A 9A9 9999 999
Contact.Input	文字列 [79]	受取人の名前。例えば、"Mr.Jones"。
Country.Input	文字列 [79]	入力为国フォーマットに選択したフォーマットを使用して国を指定します (英語名、ISO コード、または UPU コード)。有効な値の一覧は、 <a href="#">ISO 国コードとモジュール サポート</a> (735ページ) を参照してください。
FirmName.Input	文字列 [79]	会社名または企業名。
Street.Input	文字列 [79]	Street
Number.Input	建物 [79]	番号
Building.Input	文字列 [79]	建物
SubBuilding.Input	文字列 [79]	SubBuilding

応答要素	書式	説明
DeliveryService.Input	文字列 [79]	配送サービス

## 結果コード

これらの出力フィールドには、検証処理の結果に関する情報が格納されます。

表 142 : 結果コード

応答要素	結果コード
AddressType	<p>米国とカナダの住所の場合のみ、AddressType フィールドは住所のタイプを示します。次のいずれかです。</p> <p><b>F</b> 住所の妥当性が確認され社名まで修正されました。</p> <p><b>B</b> 住所の妥当性が確認され建物名まで修正されました。</p> <p><b>G</b> 住所は局留めの住所です。</p> <p><b>H</b> 住所の妥当性が確認され高層のデフォルトまで修正されました。</p> <p><b>L</b> 住所は、Large Volume Receiver (LVR) です。</p> <p><b>M</b> 住所は、軍施設の住所です。</p> <p><b>P</b> 住所の妥当性が確認され私書箱まで修正されました。</p> <p><b>R</b> 住所の妥当性が確認され地方配送路まで修正されました。</p> <p><b>S</b> 住所の妥当性が確認され通りの住所まで修正されました。</p> <p><b>U</b> タイプが不明なために住所を検証/修正できませんでした。</p>
Confidence	<p>返された住所に割り当てられた確信レベル。範囲は 0 ~ 100 です。0 は失敗を表し、100 はマッチ結果が正しいことに対する確信レベルが非常に高いことを表します。</p>
CountOverflow	<p>候補住所数が返される数より多いかどうかを示します。次のいずれかです。</p> <p><b>はい</b> 他の候補住所があります。他の候補を取得するには、 の値を増やします。</p> <p><b>いいえ</b> いいえ、これ以外の候補はありません。</p>

## 応答要素

## 結果コード

**ElementInputStatus**

**ElementInputStatus** は、入力要素を参照データに対してマッチングした際の要素ごとの情報を提供します。このフィールドの値は、バッチとパースのどちらのモードを使用するかによって変わります。このフィールドの値については、[ElementInputStatus](#)、[ElementResultStatus](#)、および [ElementRelevance](#) の解釈 (680ページ) を参照してください。

**ElementRelevance**

その土地の郵便当局の基準から基準から判断して、実際にどの住所要素が関連しているかを示します。このフィールドの値については、[ElementInputStatus](#)、[ElementResultStatus](#)、および [ElementRelevance](#) の解釈 (680ページ) を参照してください。

**ElementResultStatus**

**ElementResultStatus** は、出力フィールドが入力フィールドから変更されているか、またどのように変わっているかを示すことによって、結果を **ProcessStatus** フィールドよりも詳細に分類します。このフィールドの値については、[ElementInputStatus](#)、[ElementResultStatus](#)、および [ElementRelevance](#) の解釈 (680ページ) を参照してください。

**MailabilityScore**

郵便が住所にどの程度確実に配達されるかの予測を示します。次のいずれかです。

- 5**            確実に配達される
- 4**            ほぼ確実に配達される
- 3**            おそらく配達される
- 2**            五分五分
- 1**            配達されない可能性が高い
- 0**            配達されない

**ModeUsed**

使用された処理モードを示します。処理モードは、オプションで指定します。モードの説明については、[プロセス オプション](#) (667ページ) を参照してください。

**MultimatchCount**

住所が参照データ内の複数の候補住所と一致した場合、このフィールドには一致した候補の数が含まれます。

応答要素

結果コード

---

ProcessStatus

## 応答要素

## 結果コード

出力の品質に関する一般的な記述が含まれます。出力の品質について、詳しくは `ElementResultStatus` フィールドを参照してください。

次のいずれかです。

- V4** 検証済み。正しい入力データです。すべての要素がチェックされ、入力データが完全に一致しました。
- V3** 検証済み。入力されたデータは正しいものですが、一部またはすべての要素が正規化されたか、あるいは入力データに旧式の名前または外名 (エクソニム) が含まれます。
- V2** 検証済み。正しい入力データですが、一部の要素について、参照データが完全でないために検証できていません。
- V1** 検証済み。正しい入力データですが、ユーザによる正規化によって、正しく配達される可能性が損なわれています (たとえば選択された郵便番号の長さが短すぎるなど、ユーザによって要素の正規化が正しく行われていない)。検証によってセットされません。
- C4** 修正済み。すべての要素がチェック済みです。
- C3** 修正済みですが、一部の要素をチェックできませんでした。
- C2** 修正済みですが、配達のスレータスがわかりません (参照データがありません)。
- C1** 修正済みですが、ユーザによる正規化が正しくなかったため、配達のスレータスがわかりません。検証によってセットされません。
- I4** データを完全には修正できませんでした。かなりの確率で正しく配達されます。単一の一致が見つかりました (例えば HNO は誤りだが、参照データに HNO が 1 件だけ見つかるような場合)。
- I3** データを完全には修正できませんでした。かなりの確率で正しく配達されます。複数の一致が見つかりました (例えば HNO は誤りだが、参照データに複数の HNO が見つかるような場合)。
- I2** データを修正できませんでした。この住所で正しく配達される可能性がわずかにあります。
- I1** データを修正することができず、正しく配達される可能性もほとんどありません。
- RA** 国が強制国の設定から認識されました。
- R9** 国が `DefaultCountryISO3` の設定から認識されました。
- R8** 国が名前からエラーなしで認識されました。
- R7** 国が名前からエラー付きで認識されました。
- R6**

## 応答要素

## 結果コード

---

	国が地域から認識されました。
<b>R5</b>	国が省から認識されました。
<b>R4</b>	国が主要な都市から認識されました。
<b>R3</b>	国がフォーマットから認識されました。
<b>R2</b>	国がスクリプトから認識されました。
<b>R1</b>	国が認識されませんでした。複数の一致が見つかりました。
<b>R0</b>	国が認識されませんでした。
<b>S4</b>	完全にパースされました。
<b>S3</b>	パースされ、複数の結果が得られました。
<b>S2</b>	パースされ、エラーが発生しました。要素の位置が変わっています。
<b>S1</b>	パース エラー。入力フォーマットが一致しません。
<b>N1</b>	バリデーションエラー: 国が認識されなかったため、検証が実行されませんでした。
<b>N2</b>	バリデーションエラー: 必要な参照データベースを使用できないため、検証が実行されませんでした。
<b>N3</b>	バリデーションエラー: 国をロック解除できなかったため、検証が実行されませんでした。
<b>N4</b>	バリデーションエラー: 参照データベースが破損しているか、フォーマットが正しくないため、検証が実行されませんでした。
<b>N5</b>	バリデーションエラー: 参照データベースが古すぎるため、検証が実行されませんでした。
<b>N6</b>	バリデーションエラー: 入力データが十分でないため、検証が実行されませんでした。
<b>Q3</b>	高速実行ステータス: 提案 (完全な住所) を利用可能です。
<b>Q2</b>	高速実行ステータス: 完全な住所が提案されましたが、入力データの要素と組み合わせられています (追加または削除)。
<b>Q1</b>	高速実行ステータス: 提案された住所は完全ではありません (情報を追加してください)。
<b>Q0</b>	高速実行ステータス: 十分でない情報から提案が生成されました。

---

## 応答要素

## 結果コード

**Status**

処理試行が成功したか失敗したかをレポートします。

**NULL**

成功

**F**

失敗

**Status.Code**

失敗したものがあれば、その理由を示します。

**Status.Description**

失敗したものがあれば、その理由を説明する記述が入ります。

**ElementInputStatus**、**ElementResultStatus**、および **ElementRelevance** の解釈

**ElementInputStatus**、**ElementResultStatus**、および **ElementRelevance** の出力フィールドには、検証操作の結果を詳細に示す一連の数字が含まれます。**ElementInputStatus** にはパーシング操作に関するいくつかの情報が含まれます。

**ElementInputStatus** の値は以下のようになります。

44606040600000000060

**ElementResultStatus** の値は以下のようになります。

88F0F870F00000000040

**ElementRelevance** の値は以下のようになります。

11101010100000000000

これらのフィールドの値について理解するためには、各ポジションの数字がどの要素を表すか、および各ポジションの値の意味を知る必要があります。たとえば、先頭の数字は **PostalCode.Base** 出力フィールドの結果を示します。以下に各ポジションの意味を一覧で示します。

- ポジション 1—**PostalCode.Base**
- ポジション 2—**PostalCode.AddOn**
- ポジション 3—**City**
- ポジション 4—**Locality** および **Suburb**
- ポジション 5—**StateProvince**
- ポジション 6—**County**
- ポジション 7—**StreetName**
- ポジション 8—**SecondaryStreet**
- ポジション 9—**HouseNumber**
- ポジション 10—**Number** レベル 1



- ポジション 11—POBox
- ポジション 12—Delivery サービス レベル 1
- ポジション 13—Building レベル 0
- ポジション 14—BuildingName
- ポジション 15—Sub building レベル 0
- ポジション 16—Floor and Room
- ポジション 17—FirmName
- ポジション 18—Organization レベル 1
- ポジション 19—Country
- ポジション 20—Territory

ElementInputStatus の場合、検証の値として以下のいずれかが入ります。

- 0—空
- 1—見つからない
- 2—チェックなし (参照データなし)
- 3—誤り - 検証によってのみセット。参照データベースによれば、Number または DeliveryService のいずれかの数が正しい範囲内でない。入力はコピーされ、バッチモードでは修正されていない。インタラクティブ モードと FastCompletion では、提案が提供されている
- 4—この要素内でエラーありで一致
- 5—変更ありで一致 (挿入または削除)。例:
  - パーシング: "MainSt 1" の家番号を分割するなど
  - バリデーション: 外名 (エクソニム) が入力された場合に置換したり、フィールドのサイズを超えて入力され、国の参照データベースによって無効とされたデータを短くしたなど
- 6—エラーなしで一致

ElementInputStatus の場合、パーシングの値には以下のいずれかが入ります。

- 0—空
- 1—要素の位置を変更する必要があった
- 2—一致したが、正規化する必要があった
- 3—一致した

ElementRelevance の場合、パーシングの値には以下のいずれかが入ります。

- 0—空
- 1—要素の位置を変更する必要があった
- 2—一致したが、正規化する必要があった
- 3—一致した

ElementResultStatus の場合、(国を除くすべての住所要素に対して)以下のいずれかの値が入りません。

- 0—空
- 1—検証されず、変更されていない。元のデータがコピーされている
- 2—検証されなかったが正規化された
- 3—検証されたが、入力が無効(データベースによれば、数値が正しい範囲内でない)であるため変更されなかった入力がコピーされ、修正されていない(このステータス値はバッチモードでのみセットされる)
- 4—検証されたが、参照データが見つからないため変更されていない
- 5—検証されたが、複数の一致が見つかったため変更されていない。バッチモードでのみセットされる。そうでない場合、入力を置換する複数の提案が修正済みとしてマークされる(ステータス値 7)
- 6—検証され、入力値の除外という変更が行われている
- 7—検証され、参照データに基づく修正という変更が行われている
- 8—検証され、参照データに基づく値の追加という変更が行われている
- 9—検証され、変更はされていないが、配達ステータスがわからない(例えば、DPV の値で、指定の数値範囲が参照データと部分的にしか一致せず正しくないなど)
- C—検証され、妥当性確認済みだが名前が旧式であるため変更されている
- D—検証され、妥当性確認済みだが外名から公式の名前に変更されている
- E—検証され、妥当性確認済みだが大文字と小文字の区別、または言語に基づく正規化のため変更されている。入力が別の言語と完全に一致する場合に、検証でのみこのステータスがセットされる
- F—検証され、妥当性確認済みで、完全に一致したため変更されなかった

国(ポジション 19 と 20)に関しては、以下のいずれかの値が入ります。

- 0—空
- 1—国が認識されない
- 4—国が DefaultCountryISO3 の設定から認識された
- 5—国が認識されない。複数の一致が見つかった
- 6—国がスクリプトから認識された
- 7—国がフォーマットから認識された
- 8—国が主要な都市から認識された
- 9—国が省から認識された
- C—国が地域から認識された
- D—国がエラーありで名前から認識された
- E—国が名前からエラーなしで認識された
- F—国が ForceCountryISO3 の設定から認識された

## ValidateAddressLoqate

ValidateAddressLoqate は、郵便当局の住所データを使用して、住所を正規化し、妥当性を確認します。ValidateAddress Loqate は、情報を修正し、管轄の郵便当局が推奨する書式で住所の書式を整えることができます。また、郵便番号、都市名、州/省名など、欠落している郵便情報を追加します。

ValidateAddressLoqate は、ValidateAddressLoqate が住所の妥当性を確認したかどうか、返された住所の確信レベル、住所の妥当性が確認できなかった場合はその理由など、検証処理に関する結果インジケータも返します。

ValidateAddressLoqate は、住所のマッチングと正規化において、住所行をコンポーネントに分割し、それらを Universal Addressing モジュールの各種データベースの内容と比較します。マッチを検出した場合、入力住所をデータベース情報に合わせて正規化します。データベースにマッチしなかった場合、ValidateAddressLoqate は、オプションで入力住所の書式を整えます。書式設定プロセスでは、該当する郵便当局の規則に従って住所行の構成を試みます。

ValidateAddressLoqate は、Universal Addressing モジュールに含まれています。

### リソース URL

```
http://server:port/soap/ValidateAddressLoqate
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:val="http://www.pb.com/spectrum/services/ValidateAddressLoqate"
xmlns:spec="http://spectrum.pb.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <val:ValidateAddressLoqateRequest>
      <val:input_port>
        <val:Address>
          <val:AddressLine1>1825 Kramer Ln</val:AddressLine1>
          <val:City>Austin</val:City>
          <val:StateProvince>TX</val:StateProvince>
        </val:Address>
      </val:input_port>
    </val:ValidateAddressLoqateRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:ValidateAddressLoqateResponse
xmlns:ns2="http://spectrum.pb.com/"

xmlns:ns3="http://www.pb.com/spectrum/services/ValidateAddressLoqate">
      <ns3:output_port>
        <ns3:Address>
          <ns3:Confidence>95</ns3:Confidence>
          <ns3:CouldNotValidate/>
          <ns3:ProcessedBy>LOQATE</ns3:ProcessedBy>
          <ns3:MatchScore>100.0</ns3:MatchScore>
          <ns3:AddressLine1>1825 Kramer Ln</ns3:AddressLine1>
          <ns3:AddressLine2/>
          <ns3:City>Austin</ns3:City>
          <ns3:StateProvince>TX</ns3:StateProvince>
          <ns3:PostalCode>78758-4260</ns3:PostalCode>
          <ns3:PostalCode.Base>78758</ns3:PostalCode.Base>
          <ns3:PostalCode.AddOn>4260</ns3:PostalCode.AddOn>
          <ns3:Country>United States</ns3:Country>
          <ns3:FirmName/>
          <ns3:user_fields/>
        </ns3:Address>
      </ns3:output_port>
    </ns3:ValidateAddressLoqateResponse>
  </soap:Body>
</soap:Envelope>
```

## 要求

入力データのパラメータ

表 143 : 入力フォーマット

パラメータ	書式	説明
AddressLine1	文字列	最初の住所行。
AddressLine2	文字列	2 行目の住所行。
AddressLine3	文字列	3 行目の住所行。

パラメータ	書式	説明
AddressLine4	文字列	4 行目の住所行。
City	文字列	都市名。
Country	文字列	<p>国コードまたは名前を、以下のいずれかのフォーマットで入力します。</p> <ul style="list-style-type: none"> <li>• 2 文字の ISO 3166-1 Alpha 2 国コード</li> <li>• 3 文字の ISO 3166-1 Alpha 3 国コード</li> <li>• 英語の国名</li> </ul> <p>ISO コードの一覧については、<a href="#">ISO 国コードとモジュール サポート (735ページ)</a> を参照してください。</p>
FirmName	文字列	会社名または企業名。
PostalCode	文字列	<p>住所の郵便番号は、次のフォーマットのいずれかで表されます。</p> <p>99999 99999-9999 A9A9A9 A9A 9A9 9999 999</p>
StateProvince	文字列	州または省。

## 米国住所の住所行処理

米国住所の場合、AddressLine1 から AddressLine4 の入力フィールドの処理方法は、企業名抽出または都市化コード抽出のオプションが有効になっているかどうかによって異なります。2 つのオプションのいずれかが有効になっている場合、ValidateAddressLoqate は、4 つすべてのフィールドのデータを参照して、住所の妥当性を確認し、要求されたデータ (企業名または都市化コード) を抽出します。どちらのオプションも有効でない場合は、ValidateAddressLoqate は、空白でない最初の 2 つの住所行フィールドのみを使用して、妥当性を確認します。他の住所行フィールドのデータは、AdditionalInputData 出力フィールドに返されます。例を次に示します。

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

この住所において、企業名抽出または都市化コード抽出が有効である場合、`ValidateAddressLoqate` は、4 つすべての住所行を確認します。企業名抽出と都市化コード抽出がどちらも有効でない場合は、`ValidateAddressLoqate` は、`AddressLine1` と `AddressLine3` (空白でない最初の 2 つの住所行) を参照して、そのデータを使用して住所の妥当性を確認します。`AddressLine4` のデータは、`AdditionalInputData` 出力フィールドに返されます。

### オプション

以下の表に、`ValidateAddressLoqate` が返す情報の種類を制御するオプションの一覧を示します。

表 144 : 出力データ オプション

パラメータ	説明
<code>Database.Loqate</code>	国際住所の妥当性の確認に使用するデータベースを指定します。国際住所検証用のデータベースを指定するには、 <b>[データベース]</b> ドロップダウン リストからデータベースを選択します。
<code>OutputFieldLevelReturnCodes</code>	<p>フィールドレベルの結果インジケータを含めるかどうかを指定します。フィールドレベルの結果インジケータは、<code>ValidateAddressLoqate</code> が各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、<code>HouseNumber</code> のフィールドレベルの結果インジケータは <b>HouseNumber.Result</b> に格納されます。結果インジケータの出力フィールドの完全な一覧は、<a href="#">結果インジケータ</a> (704ページ) を参照してください。</p> <p><b>N</b>      フィールドレベルのリターンコードを出力しません(デフォルト)。</p> <p><b>Y</b>      フィールドレベルのリターンコードを出力します。</p>

## パラメータ

## 説明

## OutputFormattedOnFail

住所の妥当性を確認できない場合に書式を整えた住所を返すかどうかを指定します。住所には、その国の標準住所書式が設定されます。このオプションを選択しない場合、`ValidateAddressLoqate` が住所の妥当性を確認できないと、出力住所フィールドは空白になります。

**N** 失敗した住所の書式を整えません (デフォルト)。

**Y** 失敗した住所の書式を整えます。

**[標準住所を含める]**、**[住所行の要素を含める]**、および **[郵便情報を含める]** の各チェックボックスで指定されたフォーマットを使用して、書式を整えた住所が返されます。**[住所行の要素を含める]** を選択した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。`ValidateAddressLoqate` が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**[正規化された入力住所要素を含める]** を選択します。

このオプションをオンにする場合は、**[標準住所を含める]** と **[住所行の要素を含める]** またはそのいずれかを選択する必要があります。

**OutputRecordType** オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。**OutputRecordType=E** を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。`ValidateAddressLoqate` が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**OutputRecordType=I** を指定します。

Y を指定する場合は、**OutputRecordType** に対して "A" と "E" またはそのいずれかを指定する必要があります。

**Option.OutputRecordType** オプションで指定されたフォーマットを使用して、書式を整えた住所が返されます。

**Option.OutputRecordType=E** を指定した場合は、妥当性が確認できた住所に対しては、パース済みで妥当性を確認済みの住所が、パース済み住所要素に含まれることに注意してください。住所の妥当性が確認できなかった場合には、パース済み住所要素には、入力住所がパース済み形式で含まれることとなります。`ValidateAddressLoqate` が住所の妥当性を確認できたかどうかにかかわらず、必ず入力住所をパース済み形式で出力したい場合は、**Option.OutputRecordType=I** を指定します。

## パラメータ

## 説明

---

**OutputAddressBlocks**

実際の郵便物に印字される、書式を整えた住所を返すかどうかを指定します。住所の各行が、別々の住所ブロックフィールドに入れて返されます。**AddressBlock1** から **AddressBlock9** まで、最大 9 つの住所ブロック出力フィールドが使用されます。

例えば、以下の住所入力の場合、

AddressLine1: 4200 Parliament Place  
AddressLine2: Suite 600  
City: Lanham  
StateProvince: MD  
PostalCode: 20706

以下の住所ブロックが出力されます。

AddressBlock1: 4200 PARLIAMENT PL STE 600  
AddressBlock2: LANHAM MD 20706-1882  
AddressBlock3: UNITED STATES OF AMERICA

**ValidateAddressLoqate** は、郵便当局の規格に従って住所の書式を整え、住所ブロックの形式にします。国名は、万国郵便連合 (UPU) の国名で返されます。オプションは、住所ブロックの国名には影響を与えないことに注意してください。[国フォーマット] オプションは、**Country** 出力フィールドに返される名前だけに影響を与えます。

次のいずれかです。

---



## パラメータ

## 説明

## AmasFormatting

Address Matching Approval System (AMAS) 表記を使用して出力住所データをフォーマットすることを指定します。

このオプションを使用すると、Validate Address Loqate は、住所を正規化するとき AMAS ルールを使用するようになります。AMAS は、オーストラリア郵政公社が定める、住所規格を徹底するためのプログラムです。AMAS 書式設定表記の詳細については、『Address Matching Approval System (AMAS) Handbook』を参照してください。

このオプションを使用すると、出力データは次のように変更されません。

- 数値フィールドにはゼロが付加されます。この影響を受けるのは、HouseNumber、HouseNumber2、PostalDeliveryNumber、および DPID の各出力フィールドです。例えば、入力フィールドが 298 New South Head Rd Double Bay NSW 2028 の場合、HouseNumber フィールドの形式は 298 から 00298 に変更されます。
- 一致しない場合、DPID フィールドの桁はすべてゼロになります。例えば、00000000 などです。
- 一致しない場合、すべてゼロを含む数値フィールドを除き、すべてのリターン フィールド (パース済み住所要素) が空白になります。
- CCD フィールドは出力されません。

有効な値は、次のとおりです。

**N** AMAS 表記を使用して出力データをフォーマットしません (デフォルト)。

**Y** AMAS 表記を使用して出力データをフォーマットします。

注：このオプションを選択すると、[許容レベル] フィールドと [最小マッチ スコア] フィールドの選択に関わらず、AMAS フォーマットで結果が返されます。

## OutputCasing

出力データの大文字と小文字の区別を指定します。次のいずれかです。

**M** 出力には、大文字と小文字が混在させます (デフォルト)。例:  
123 Main St  
Mytown FL 12345

**U** 出力に大文字を使用します。例:  
123 MAIN ST  
MYTOWN FL 12345

## パラメータ

## 説明

## HomeCountry

デフォルト国を指定します。大部分の住所が存在する国を指定してください。例えば、処理する住所の大部分がドイツにある場合は、ドイツを指定します。ValidateAddressLoqate は、[StateProvince]、[PostalCode]、および [Country] の各住所フィールドから国を特定できなかった場合、指定された国を使用して、住所の検証を試みます。有効な国名には次のものがあります。

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

パラメータ	説明
OutputCountryFormat	<p><b>Country</b> 出力フィールドに返される国名に使用するフォーマットを指定します。例えば、英語を選択した場合、"Deutschland" という国名は "Germany" として返されます。</p> <p><b>E</b> 英語の国名を使用します (デフォルト)。</p> <p><b>I</b> 国名の代わりにその国の 2 文字の ISO の略語を使用します。</p> <p><b>U</b> 国名の代わりにその国の万国郵便連合 (Universal Postal Union: UPU) の略語を使用します。</p>
OutputScript	<p>出力がどのアルファベットまたはスクリプトで返されるかを指定します。このオプションは双方向で、通常はネイティブからラテン文字へ、およびラテン文字からネイティブへ実行されます。</p> <p><b>Input</b> 書き直しを実行せず、入力と同じスクリプトで出力します (デフォルト)。</p> <p><b>Native</b> 使用可能な場合は、選択した国のネイティブスクリプトで出力します。</p> <p><b>Latn</b> 英語の値を使用します。</p>

パラメータ

説明

---

許容レベル

AcceptanceLevel

## パラメータ

## 説明

レコードが正常に処理されたとみなされるために達成する必要のある、最小検証レベルを指定します。このフィールドの値は、“処理後検証マッチ レベル” と呼ばれる、Address Verification Code の 2 つめの文字に対応します。

- **5** — 配達ポイント (建物または郵便受け)。入力レコードの ApartmentNumber、HouseNumber、Street、City、StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。ApartmentNumber は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、ApartmentNumber は他のフィールドよりも細かいレベルであるため、Loqate エンジンはこの場合に、ApartmentNumber を特定できるはずでず。Loqate エンジンが ApartmentNumber とその他のフィールドをパーシングできない場合は、確信レベルは 0 となります。
- **4** — 敷地または建物。入力レコードの HouseNumber、Street、City、StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。HouseNumber は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、HouseNumber は他のフィールドよりも細かいレベルであるため、Loqate エンジンはこの場合に、HouseNumber を特定できるはずでず。Loqate エンジンが HouseNumber とその他のフィールドをパーシングできない場合は、確信レベルは 0 となります。
- **3** — 大通り、道路、または通り。入力レコードの Street、City、StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。City は正しいが StateProvince が正しくない場合、確信レベルは中程度になりますが、City は StateProvince に含まれるため、Loqate エンジンはこの場合に、StateProvince を特定できるはずでず。Loqate エンジンが City または両方のフィールド (City と StateProvince) をパーシングできない場合は、確信レベルは 0 となります。
- **2** — 地方 (都市または町)。入力レコードの City と StateProvince の両方が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。City は正しいが StateProvince が正しくない場合、確信レベルは中程度になりますが、City は StateProvince に含まれるため、Loqate エンジンはこの場合に、StateProvince を特定できるはずでず。Loqate エンジンが City または両方のフィールド (City と StateProvince) をパーシングできない場合は、確信レベルは 0 となります。
- **1** — 行政区域 (州または地域)。入力レコードの StateProvince が、Loqate のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。
- **0** — なし。これは、最も制限の緩いマッチオプションと同じです。

パラメータ	説明
IsDuplicateHandlingMaskEnable	<p>重複処理マスクを有効にし、重複レコードの処理および削除の方法を指定します。次のオプションから 1 つ以上を選択します。</p> <p><b>S</b> デフォルトで選択されています。入力の前処理により、単一フィールドで発生している重複を削除します。</p> <p><b>C</b> デフォルトで選択されています。入力の前処理により、すべてのフィールドわたり重複を削除します。</p> <p><b>T</b> 入力の前処理により、標準住所フィールドでないフィールド内の重複を削除します。</p> <p><b>F</b> デフォルトで選択されています。検証の出力の後処理により、検証されていないフィールドから重複を削除します。</p>
MinimumMatchScore	<p>Validate Address Loqate において、Loqate 参照データベースでマッチ結果を得るために、住所に加える変更の度合いを、0～100 の間の数値で指定します。数値が小さいほど、大きな変更が許容されます。100 の値は、パーシング後に入力住所と検証済み住所がほぼ同一であることを意味します。0 の値は、検証済み住所を得るためにパーシング後の入力住所を大きく変更してしまってもよいことを意味します。</p>
KeepMultimatch	<p>一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを指定します。</p> <p><b>Y</b> 複数のマッチを返します (デフォルト)。</p> <p><b>N</b> 複数のマッチを返しません。</p> <p>詳細については、<a href="#">複数マッチを返す</a> (694 ページ) を参照してください。</p>
FailMultipleMatches	<p>一致する可能性のある住所が複数存在する入力住所に対して、複数の住所を許可しません。</p>

### 複数マッチを返す

ValidateAddressLoqate が、入力住所に一致する可能性のある複数の住所を郵便データベース内で検出した場合に、ValidateAddressLoqate がそれらの複数の住所を返すように設定できます。例えば、次の住所は米国郵便データベース内の複数の住所にマッチします。

PO BOX 1 New York, NY

## オプション

複数マッチを返すには、次の表に示すオプションを使用します。

表 145 : 複数マッチのオプション

### 説明/有効値

一致する可能性のある住所を複数持つ入力住所に対して複数の住所を返すかどうかを示します。

返す住所の最大数を示す 1 ~ 10 の数字を入力します。デフォルト値は 1 です。

注：との違いは、は複数マッチによって失敗が返され、は複数マッチによって 1 つのレコードが返される点です。

どの出力住所が候補住所かを特定するには、必要があります。このように設定すると、候補住所のレコードのフィールドレベルの結果インジケータに 1 つ以上の値 "M" が格納されます。

## 出力

複数マッチを返すよう選択した場合、住所は指定した住所フォーマットで返されます。住所フォーマットの指定については、[オプション](#) (686ページ) を参照してください。どのレコードが候補住所であるかを特定するには、フィールドレベルの結果インジケータに複数の値 "M" があるかどうかを調べます。詳細については、[結果インジケータ](#) (704ページ) を参照してください。

### マッチ スコアのしきい値オプション

マッチ スコアのしきい値を設定するための 2 つのオプションがあります。

注：これらのオプションは `Validate Address Loqate` のユーザ インターフェイスにはなく、以下のファイルの中にあります。

```
SpectrumDirectory/server/modules/loqate/env.properties
```

**[MatchScoreAbsoluteThreshold]** オプションは、レコードがマッチングの候補とみなされるために達しなければならない最小マッチ スコアを指定するために使用されます。デフォルト値は 60 で、最大値は 100 です。

**[MatchScoreThresholdFactor]** は、最も高いマッチング結果を 100 とした場合の係数を表す値です。この値は、結果候補を検討する際のボーダーラインとして使用されます。係数の値が高いほど、良い検証結果が得られる確率が高くなります。デフォルト値は 95 で、最大値は 100 です。

## 応答

ValidateAddressLoqate からの出力には、選択した出力カテゴリに応じてさまざまな情報が含まれます。

### 標準住所出力

標準住所出力は、宛名ラベルに表記される住所に対応する 4 行の住所で構成されます。都市、州/省、郵便番号などのデータも、標準住所出力に含まれます。**[標準住所を含める]** チェックボックスを選択 **OutputRecordType = A** と設定した場合、ValidateAddressLoqate は、妥当性を確認した住所に対し、標準住所出力を返します。したかどうにかかわらず、妥当性が確認できなかった住所に対しては、標準住所フィールドが必ず返されます。妥当性が確認されなかった住所に対しては、標準住所出力フィールドには、入力住所がそのまま含まれます ("パス スルー" データ)。ValidateAddressLoqate において、妥当性が確認できなかった場合に、郵便当局の規格に従って住所を正規化するには、します。

表 146 : 標準住所出力

応答要素	説明
AdditionalInputData	特定の住所コンポーネントに一致しなかった入力データ。詳細については、 <a href="#">その他の入力データについて</a> を参照してください。
AddressLine1-4	住所の妥当性が確認された場合は、妥当性が確認され、正規化された住所の 1 行目です。住所の妥当性が確認できなかった場合は、入力住所の 1 行目がそのまま出力されます。住所ブロック出力フィールドは最大で、AddressLine1 から AddressLine4 の 4 つになります。
City	妥当性が確認された都市名。
Country	で選択した、以下のいずれかのフォーマットで示された国。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>



応答要素	説明
FirmName	妥当性が確認された企業名。
PostalCode	妥当性が確認された ZIP Code™ または郵便番号。
PostalCode.AddOn	ZIP Code™ の 4 桁のアドオン部分。例えば、60655-1844 という ZIP Code™ において、4 桁のアドオン部分は 1844 になります
PostalCode.Base	5 桁の ZIP Code™。例: 20706。
StateProvince	妥当性が確認された州または省の略称。

#### パース済み住所要素出力

した場合、出力住所は、パース済み住所の形式で書式設定されます。ValidateAddressLoqate で、妥当性が確認できなかった場合に、パース済み住所形式で書式設定されたデータ (正規化済み住所) を返すには、します。

注: ValidateAddressLoqate で、妥当性が確認できたかどうかにかかわらず、常にパースした入力データを返すには、します。詳細については、[パース済み入力 \(700ページ\)](#) を参照してください。

表 147 : パース済み住所出力

Response Element	説明
AddressBlock1-9	<p>AddressBlock 出力フィールドには、正規化済み、または標準化済みの住所が、実際の郵便物に印刷される形式にフォーマットされて入ります。Validate Address Global は住所を郵便当局の規格に従って住所ブロックにフォーマットします。住所の各行が、別々の住所ブロックフィールドに入れて返されます。AddressBlock1 から AddressBlock9 まで、最大 9 つの住所ブロック出力フィールドが使用されます。例えば、以下の入力住所の場合、</p> <p>AddressLine1: 4200 Parliament Place AddressLine2: Suite 600 City: Lanham StateProvince: MD PostalCode: 20706</p> <p>以下の住所ブロックが出力されます。</p> <p>AddressBlock1: 4200 PARLIAMENT PL STE 600 AddressBlock2: LANHAM MD 20706-1882</p>
ApartmentLabel	<p>アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b></p>
ApartmentNumber	<p>アパート番号。例: 123 E Main St <b>APT 3</b></p>
ApartmentNumber2	<p>補助的なアパート番号。例: 123 E Main St <b>APT 3, 4th Floor</b></p> <p>注: このリリースでは、このフィールドは常に空白になります。</p>
Building	<p>個々の場所を識別するためのわかりやすい名前。</p>
City	<p>妥当性が確認された都市名</p>
Country	<p>国。フォーマットは、 で選択したものになります。</p> <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>

Response Element	説明
County*	国における最小の地理的データ要素。例: 米国の郡
FirmName	妥当性が確認された企業名
HouseNumber	家番号 1。例: <b>123 E Main St Apt 3</b>
LeadingDirectional	接頭方向指示。例: <b>123 E Main St Apt 3</b>
POBox	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode	妥当性が検証された郵便番号。米国住所に対しては、ZIP Code と呼びます。
Principality *	国における最大の地理的データ要素
StateProvince	妥当性が確認された州または省の名前
StreetAlias	通り名のエイリアス。通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。例: <b>123 E Main St Apt 3</b>
StreetName	通り名。例: <b>123 E Main St Apt 3</b>
StreetSuffix	通り接尾語。例: <b>123 E Main St Apt 3</b>
Subcity*	[Locality] フィールドの内容に依存する、より小さな居留区データ要素。例: トルコ人居留区。

Response Element	説明
Substreet*	国における従属通りまたはブロック データ要素。例: 英国の従属通り。
TrailingDirectional	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>

\*これはサブフィールドであり、データを含まない場合があります。

### パース済み入力

出力には、パース済み形式で入力住所を含めることができます。このようなタイプの出力は、"パース済み入力" と呼ばれます。パース済み入力フィールドには、**ValidateAddress** が住所の妥当性を検証したかどうかにかかわらず、入力として使用される住所データが含まれます。パース済み入力は、住所の妥当性を検証できた場合にパース済み住所要素に妥当性が検証された住所が含まれ、オプションで、住所の妥当性が検証できなかった場合には入力データが含まれるという点で、"パース済み住所要素" 出力と異なります。パース済み入力には、**ValidateAddress** が住所の妥当性を検証したかどうかにかかわらず、常に入力住所が含まれます。

パース済み入力フィールドを出力に含めるには、。

表 148 : パース済み入力

応答要素	説明
ApartmentLabel.Input	アパート指定子 (STE や APT など)。例: 123 E Main St <b>APT 3</b>
ApartmentNumber.Input	アパート番号。例: 123 E Main St <b>APT 3</b>
City.Input	妥当性が確認された都市名
Country.Input	国。フォーマットは、 で選択したものになります。 <ul style="list-style-type: none"> <li>• ISO コード</li> <li>• UPU コード</li> <li>• 英語</li> </ul>

応答要素	説明
County.Input*	国における最小の地理的データ要素。例: 米国の郡
FirmName.Input	妥当性が確認された企業名
HouseNumber.Input	家番号 1。例: <b>123</b> E Main St Apt 3
LeadingDirectional.Input	接頭方向指示。例: 123 <b>E</b> Main St Apt 3
POBox.Input	私書箱番号。住所が地方配送路住所である場合は、地方配送路の私書箱番号がここに表示されます。
PostalCode.Input	妥当性が検証された郵便番号。米国住所に対しては、ZIP Code と呼びます。
Principality.Input *	国における最大の地理的データ要素
StateProvince.Input	妥当性が確認された州または省の名前
StreetAlias.Input	通り名のエイリアス。通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。基本名とは、通り全体に適用される名称です。例えば、StreetName が "N MAIN ST" の場合、StreetAlias フィールドには "MAIN" が含まれ、StreetSuffix フィールドには大通りタイプである "ST" が返されます。
StreetName.Input	通り名。例: 123 <b>E Main St</b> Apt 3
StreetSuffix.Input	通り接尾語。例: 123 E Main St <b>Apt 3</b>

応答要素	説明
Subcity.Input*	[Locality] フィールドの内容に依存する、より小さな居留区データ要素。例: トルコ人居留区。
Substreet.Input*	国における従属通りまたはブロック データ要素。例: 英国の従属通り。
TrailingDirectional.Input	接尾方向指示。例: 123 Pennsylvania Ave <b>NW</b>

\*これはサブフィールドであり、データを含まない場合があります。

#### ジオコード出力

`ValidateAddressLoqate` は、緯度/経度、ジオコーディング マッチ コード、従属する地方、従属地方に含まれる地方、従属する大ストリート、下位行政区画と上位行政区画、検索距離を出力として返します。マッチ コードは、入力住所がどの程度まで既知の住所に一致したかを表すとともに、マッチングの全体的なステータスを示します。検索距離コードは、ジオコードが住所の実際の物理的な位置にどの程度近いかを表します。

表 149 : ジオコード住所出力

応答要素	説明
Geocode.MatchCode	<p>住所に対するジオコード マッチングのステータスとレベルを表す 2 バイト コード。</p> <p>最初のバイトはジオコーディング ステータスを表し、次のいずれかになります。</p> <p><b>A</b> 入力住所に一致する複数のジオコード候補が検出され、それらの平均が返されました。</p> <p><b>I</b> ある範囲の入力住所のロケーションからジオコードを補間することができました。</p> <p><b>P</b> 入力住所に一致する単一のジオコードが検出されました。</p> <p><b>U</b> 入力住所に対するジオコードを生成できませんでした。</p> <p>2 つめのバイトはジオコード マッチングのレベルを表し、次のいずれかになります。</p> <p><b>5</b> 配達ポイント (郵便受けまたは従属する建物)</p> <p><b>4</b> 敷地または建物</p> <p><b>3</b> Thoroughfare</p> <p><b>2</b> Locality</p> <p><b>1</b> 行政区画</p> <p><b>0</b> なし</p>
Latitude	小数第 5 位まで計算される 8 桁の度数 (指定したフォーマットで表記されます)。
Longitude	小数第 5 位まで計算される 8 桁の度数 (指定したフォーマットで表記されます)。
SearchDistance	メートル単位で表した精度の半径。与えられたジオコードと実際の物理的位置との間の推測最大距離を表します。このフィールドは、基盤の参照データから導き出され、その精度と対象範囲に依存します。

表 150 : 都市/通り/郵便番号セントロイド マッチ コード

要素	マッチ コード
住所ポイント	P4
住所ポイント補間済み	I4
通りセントロイド	A4/P3
郵便番号/都市セントロイド	A3/P2/A2

注 : `Geocode.Match.Code` は、通り セグメントに対して 2 つの座標を返すことはしません (通りの一部の開始と終了など)。代わりに、入力に対するリターンコードが I3 になった (大通りまたは通り レベルに補間され、敷地番号は入力されていない) 場合は、通り全体が計算に使用されます。

#### 結果インジケータ

結果インジケータは、住所に対して実行した処理の種類に関する情報を提供します。結果インジケータには、次の 2 種類があります。

#### レコード レベルの結果インジケータ

レコード レベルの結果インジケータは、各レコードに対する `ValidateAddressLoqate` 処理の結果に関するデータを提供します。例えば、マッチングの成功または失敗、住所を処理したコーダーなどの詳細情報を示します。以下の表に、`ValidateAddressLoqate` が返すレコード レベルの結果インジケータの一覧を示します。

表 151 : レコード レベル インジケータ

応答要素	説明
Confidence	返された住所に割り当てられた確信レベル。範囲は 0 ~ 100 です。0 は失敗を表し、100 はマッチ結果が正しいことに対する確信レベルが非常に高いことを表します。複数のマッチ結果がある場合、確信レベルは 0 です。この数値の計算方法については、 <a href="#">Validate Address Loqate 確信アルゴリズムの概要</a> を参照してください。



## 応答要素

## 説明

**CouldNotValidate**

マッチしなかった場合の、妥当性が確認できなかった住所コンポーネント。

- ApartmentNumber
- HouseNumber
- StreetName
- PostalCode
- City
- Directional
- StreetSuffix
- Firm
- POBoxNumber

注：複数のコンポーネントがカンマ区切りリストとして返されることがあります。

**MatchScore**

**MatchScore** は、入力データと、最も近い一致として検出された参照データとの間の類似性を表します。**MatchScore** は、一致を得るために入力住所をどれだけ変更したかを表す **Confidence** とはまったく異なるものです。**MatchScore** の意味は、米国住所と米国以外の住所で異なります。

`int getFieldMatchscore (unit record, const char*)` 関数は、入力データと、最も近い一致として検出された参照データとの間の類似性を表す 0 ~ 100 の整数値です。100 は、入力データに、エイリアス、大文字小文字、付加記号以外の変更を加えなかったことを表します。0 は、入力データと、最も近い一致として検出された参照データとの間にまったく類似性がないことを表します。

注：**Validate Address Loqate** および **Advanced Matching** モジュールのコンポーネントは、どちらも **MatchScore** フィールドを使用します。データフローの出力の **MatchScore** フィールドの値は、出力ステージに送られる前に最後に値を変更したステージによって決まります。データフローに **Validate Address Loqate** および **Advanced Matching** モジュールのコンポーネントが含まれ、各ステージの **MatchScore** 出力フィールドを確認したい場合は、**Transformer** ステージを使用して、**MatchScore** 値を他のフィールドにコピーしてください。例えば、**Validate Address Loqate** によって **MatchScore** という出力フィールドが作成され、**Transformer** ステージによって **Validate Address Loqate** の **MatchScore** フィールドが **AddressMatchScore** というフィールドにコピーされます。マッチャーステージを実行すると、マッチャーから得た値が **MatchScore** フィールドに設定され、**Validate Address Loqate** から得た **AddressMatchScore** の値が引き渡されます。

**ProcessedBy**

住所を処理した住所コーダーです。

**LOQATE**

Loqate コーダーが住所を処理しました。

応答要素	説明
Status	<p>マッチの成功または失敗。複数のマッチがある場合、一致する可能性のあるすべてのものに対してこのフィールドが "F" になります。</p> <p><b>NULL</b>                      成功</p> <p><b>F</b>                              失敗</p>
Status.Code	<p>失敗の原因 (ある場合)。</p> <ul style="list-style-type: none"> <li>• UnableToValidate</li> </ul>
Status.Description	<p>問題の説明 (ある場合)。</p> <p><b>Address Not Found</b>      Status.Code=UnableToValidate の場合にこの値が表示されます。</p>

#### フィールドレベルの結果インジケータ

フィールドレベルの結果インジケータは、`ValidateAddressLoqate` が各住所要素をどのように処理したかを示します。フィールドレベルの結果インジケータは、修飾子 "Result" で返されます。例えば、`HouseNumber` のフィールドレベルの結果インジケータは **HouseNumber.Result** に格納されます。

フィールドレベルの結果インジケータを有効にするには、。

次の表に、フィールドレベルの結果インジケータの一覧を示します。特定のフィールドが住所に適用されない場合、結果インジケータが空白になる場合があります。

表 152 : フィールドレベルの結果インジケータ

応答要素	説明
ApartmentLabel.Result	<b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
	<b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。
	<b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。
	<b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
	<b>R</b> アパート ラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
	<b>S</b> 正規化。このオプションには、標準の略語が含まれます。
	<b>U</b> マッチしない。カナダの住所には適用されません。
<b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。	
ApartmentNumber.Result	<b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
	<b>C</b> 修正済み。カナダの住所のみ。
	<b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
	<b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国EWSにマッチする米国の住所には、Pの値が割り当てられます。米国およびカナダの住所のみをサポートします。
	<b>R</b> アパート番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
	<b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
	<b>U</b> マッチしない。
<b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。	

応答要素	説明
City.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> ハイフンの欠落または句読文字エラー。カナダの住所のみ。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。</p> <p><b>R</b> 都市名が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。</p> <p><b>U</b> マッチしない。カナダの住所には適用されません。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
Country.Result	<p>これらの結果コードは、米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
County.Result*	<p>国における最小の地理的データ要素。例: 米国の郡</p>

応答要素	説明
FirmName.Result	<b>C</b> 修正済み。米国住所にのみ適用されます。
	<b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
	<b>U</b> マッチしない。米国およびカナダの住所のみをサポートします。
	<b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。米国住所にのみ適用されます。
HouseNumber.Result	<b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。
	<b>C</b> 修正済み。カナダの住所のみ。
	<b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
	<b>O</b> 範囲外。米国またはカナダの住所には適用されません。
	<b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
	<b>R</b> 家番号が必須ですが、入力住所から欠落しています。カナダの住所のみ。
	<b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。
	<b>U</b> マッチしない。
	<b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

応答要素	説明
<b>LeadingDirectional.Result</b>	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。空白のない入力、空白のない値に修正されました。米国住所にのみ適用されます。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした、出力に保持されました。カナダの住所のみ。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。カナダの住所には適用されません。</p>
<b>POBox.Result</b>	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。</p> <p><b>C</b> 修正済み。カナダの住所のみ。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数マッチ。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした、出力に保持されました。カナダの住所のみ。</p> <p><b>R</b> 私書箱番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>

応答要素	説明
PostalCode.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。カナダの住所には適用されません。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国住所には適用されません。</p> <p><b>R</b> 郵便番号が必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国またはカナダの住所には適用されません。</p> <p><b>U</b> マッチしない。例えば、通り名と郵便番号が一致しない場合、StreetName.Result と PostalCode.Result の両方に U が割り当てられます。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
PostalCode.Type	<p><b>P</b> ZIP Code™ には、PO Box 住所のみが含まれます。米国住所にのみ適用されます。</p> <p><b>U</b> ZIP Code™ は、特定の会社または場所に割り当てられたユニークな ZIP Code™ です。米国住所にのみ適用されます。</p> <p><b>M</b> ZIP Code™ は、軍施設の住所です。米国住所にのみ適用されます。</p> <p><b>NULL</b> ZIP Code™ は、標準 ZIP Code™ です。</p>
Principality.Result *	国における最大の地理的データ要素

## 応答要素

## 説明

**StateProvince.Result**

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国住所にのみ適用されます。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国またはカナダの住所には適用されません。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国およびカナダの住所のみをサポートします。
- R** アパート ラベルが必須ですが、入力住所から欠落しています。米国住所にのみ適用されます。
- S** 正規化。このオプションには、標準の略語が含まれます。米国住所には適用されません。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

**StreetAlias.Result**

通り名のエイリアス。通常は、通り上の特定の範囲の住所のみに対して使用されます。通り名のエイリアスを出力に使用しないと指定した場合は、通りにエイリアス名があるかどうかにかかわらず、出力では通りの "基本" 名が使用されます。基本名とは、通り全体に適用される名称です。例えば、**StreetName** が "N MAIN ST" の場合、**StreetAlias** フィールドには "MAIN" が含まれ、**StreetSuffix** フィールドには大通りタイプである "ST" が返されます。



応答要素	説明
StreetName.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。カナダの住所のみ。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。米国住所には適用されません。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。米国およびカナダの住所のみをサポートします。</p> <p><b>U</b> マッチしない。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
StreetSuffix.Result	<p><b>A</b> 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。</p> <p><b>C</b> 修正済み。米国およびカナダの住所のみをサポートします。</p> <p><b>F</b> 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。</p> <p><b>M</b> 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。</p> <p><b>P</b> パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。</p> <p><b>S</b> 正規化。このオプションには、標準の略語が含まれます。</p> <p><b>U</b> マッチしない。米国住所には適用されません。</p> <p><b>V</b> 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。</p>
Subcity.Result*	[Locality] フィールドの内容に依存する、より小さな居留区データ要素。例: トルコ 人居留区。

## 応答要素

## 説明

Substreet.Result\*

国における従属通りまたはブロック データ要素。例: 英国の従属通り。

TrailingDirectional.Result

- A** 追加済み。フィールドが空白の入力フィールドに追加されました。米国およびカナダの住所のみをサポートします。
- C** 修正済み。米国およびカナダの住所のみをサポートします。
- F** 書式設定済み。郵便規格に準拠するよう、スペースや句読文字が変更されました。米国またはカナダの住所には適用されません。
- M** 複数。入力住所が郵便データベース内の複数レコードにマッチし、マッチする各レコードのこのフィールドの値が異なります。米国住所にのみ適用されます。
- P** パススルー。データは検証プロセスで使用されませんでした。出力に保持されました。カナダの住所のみ。
- S** 正規化。このオプションには、標準の略語が含まれます。
- U** マッチしない。カナダの住所には適用されません。
- V** 妥当性が確認されました。データは正しいことが確認され、入力から変更されていません。

\*これはサブフィールドであり、データを含まない場合があります。

## AVC コード

Address Verification Code (AVC) は、住所に対する精度インジケータで構成される 11 バイトのコードです。これらのコードは、処理結果の品質を表し、必要に応じて入力データを修正する方法に関するガイドラインを示します。個別住所のそれぞれに、独自のコードが付与されます。このコードは、データフローの出力内で自動的に返されます。AVC は、以下のような形式です。

V44-I44-P6-100

AVC は、8 つの部分で構成されます。

- 検証ステータス
- 処理後検証マッチ レベル
- 処理前検証マッチ レベル
- パーシング ステータス
- 辞書識別マッチ レベル
- コンテキスト識別マッチ レベル
- 郵便番号ステータス

- マッチスコア

### 検証ステータス

住所の検証レベル。

- **V** — 検証済み。入力データは、使用可能な参照データからの単一のレコードと完全に一致しています。シンプルな住所検証では、このコードが返されることが最良の結果とみなされます。
- **P** — 部分的に検証済み。入力データは、使用可能な参照データからの単一のレコードと部分的に一致しています。住所情報の詳細なデータが提供されているが、完全な検証を行うためには追加情報が必要であることを意味する可能性があります。
- **A** — 曖昧。入力に一致する可能性のある複数の住所があります。
- **U** — 検証不可。住所を検証するための十分な情報がない場合や、入力クエリが読み取れない場合に、これが返されます。出力フィールドには入力データが含まれます。
- **R** — 元に戻されました。レコードを、指定された最小許容レベルで検証できませんでした。元に戻すための最小レベルなどの詳細設定オプションが、処理において設定されている場合に生じます。出力フィールドには入力データが含まれます。
- **C** — 矛盾。相反する値を持つ複数の参照データと近似一致します。

### 処理後検証マッチ レベル

使用可能な参照データに対する、入力データの処理後のマッチ レベル。

- **5** — 配達ポイント (建物または郵便受け)。入力レコードの `ApartmentNumber`、`HouseNumber`、`Street`、`City`、`StateProvince` が、`Loqate` のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。`ApartmentNumber` は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、`ApartmentNumber` は他のフィールドよりも細かいレベルであるため、`Loqate` エンジンはこの場合に、`ApartmentNumber` を特定できるはずですが、`Loqate` エンジンが `ApartmentNumber` とその他のフィールドをパーシングできない場合は、確信レベルは **0** となります。
- **4** — 敷地または建物。入力レコードの `HouseNumber`、`Street`、`City`、`StateProvince` が、`Loqate` のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。`HouseNumber` は正しいがその他のフィールドが正しくない場合、確信レベルは中程度になりますが、`HouseNumber` は他のフィールドよりも細かいレベルであるため、`Loqate` エンジンはこの場合に、`HouseNumber` を特定できるはずですが、`Loqate` エンジンが `HouseNumber` とその他のフィールドをパーシングできない場合は、確信レベルは **0** となります。
- **3** — 大通り、道路、または通り。入力レコードの `Street`、`City`、`StateProvince` が、`Loqate` のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。`City` は正しいが `StateProvince` が正しくない場合、確信レベルは中程度になりますが、`City` は `StateProvince` に含まれるため、`Loqate` エンジンはこの場合に、`StateProvince` を特定できるはずですが、`Loqate` エンジンが `City` または両方のフィールド (`City` と `StateProvince`) をパーシングできない場合は、確信レベルは **0** となります。

- **2** — 地方 (都市または町)。入力レコードの **City** と **StateProvince** の両方が、**Loqate** のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。**City** は正しいが **StateProvince** が正しくない場合、確信レベルは中程度になりますが、**City** は **StateProvince** に含まれるため、**Loqate** エンジンはこの場合に、**StateProvince** を特定できるはずですが、**Loqate** エンジンが **City** または両方のフィールド (**City** と **StateProvince**) をパーシングできない場合は、確信レベルは **0** となります。
- **1** — 行政区域 (州または地域)。入力レコードの **StateProvince** が、**Loqate** のリファレンス データセットと一致する場合、レコードは引き渡されるか、高い確信レベルを付与されます。
- **0** — なし。これは、最も制限の緩いマッチ オプションと同じです。

### 処理前検証マッチ レベル

使用可能な参照データに対する、入力データの処理前のマッチ レベル。

- **5** — 配達ポイント (建物または郵便受け)
- **4** — 敷地または建物。
- **3** — 大通り、道路、または通り。
- **2** — 地方 (都市または町)。
- **1** — 行政区域 (州または地域)。
- **0** — なし。

### パーシング ステータス

住所のパーシング レベル。

- **I** — 識別およびパーシング済み。入力データは識別され、各コンポーネントに配置されました。例えば、"123 Kingston Av" に対して **Validate Address Loqate** は、"123" が敷地番号で、"Kingston" が大通り名で、"Av" または "Avenue" が大通りのタイプであると判断できます。
- **U** — パーシング不可。**Validate Address Loqate** は、入力データを識別およびパーシングできませんでした。"未検証"の検証ステータスと同様に、入力データは不完全またはあいまいでした。

### 辞書識別マッチ レベル

パターン マッチング (例えば、数値は敷地番号である可能性があります) や辞書マッチング (例えば、"rd" は大通りのタイプ "road" であったり、"London" は地方であったりする可能性があります) の適用による、入力データの認識済み形式のレベル。

- **5** — 配達ポイント (建物または郵便受け)
- **4** — 敷地または建物。
- **3** — 大通り、道路、または通り。
- **2** — 地方 (都市または町)。
- **1** — 行政区域 (州または地域)。
- **0** — なし。

### コンテキスト識別マッチ レベル

出現のコンテキストに基づいて認識可能な、入力データのレベル。最も精度の低い形式のマッチングで、単語を特定の住所要素として識別することに基づいて行われます。例えば、前に敷地らしい要素があり、後に地方らしい要素が続き、後の項目は参照データまたは辞書とのマッチングによって識別されている場合、入力は大通りであると判断できる可能性があります。

- **5** — 配達ポイント (建物または郵便受け)
- **4** — 敷地または建物。
- **3** — 大通り、道路、または通り。
- **2** — 地方 (都市または町)。
- **1** — 行政区域 (州または地域)。
- **0** — なし。

### 郵便番号ステータス

郵便番号の検証レベル。

- **P8** — PostalCodePrimary と PostalCodeSecondary が検証済みです。
- **P7** — PostalCodePrimary は検証済みで、PostalCodeSecondary は追加または変更されています。
- **P6** — PostalCodePrimary が検証済みです。
- **P5** — PostalCodePrimary が、小さな変更を加えることによって検証済みです。
- **P4** — PostalCodePrimary が、大きな変更を加えることによって検証済みです。
- **P3** — PostalCodePrimary が追加されています。
- **P2** — PostalCodePrimary が辞書によって識別されています。
- **P1** — PostalCodePrimary がコンテキストによって識別されています。
- **P0** — PostalCodePrimary が空です。

### マッチ スコア

識別済みの入力データと、レコードの出力データの間類似性を表す 0 ~ 100 の間の数値。100 は、入力データに追加、エイリアス、大文字小文字、付加記号以外の変更を加えなかったことを表します。0 は、入力データと出力データの間まったく類似性がないことを表します。

### AMAS 出力

次の表に、ValidateAddressAUS が出力する標準フィールドを示します。

表 153 : 出力フィールド

応答要素	説明
Barcode	DPID に基づく標準バーコード。 <b>F</b> 失敗 (バーコード不検出) <b>20 桁の数字</b> 成功
DPID	配達ポイント識別子。通り住所などの郵便物配達ポイントを一意に識別する 8 桁の数字。オーストラリア郵政公社郵便住所ファイルに規定されています。  注：このフィールドは、AMAS で検証されていないオーストラリアの住所の場合は "00000000" となり、オーストラリア以外の住所の場合は空になります。
FloorNumber	フロアルevel番号。例: 123 E Main St Apt 3, <b>4th Floor</b>
FloorType	フロアルevel タイプ。例: 123 E Main St Apt 3, <b>4th Floor</b>
PostalBoxNum	郵便配達番号。例: PO Box 42

## Universal Name モジュール

### OpenNameParser

OpenNameParser は、名前データフィールドにある個人名、企業名、またはその他の名称を構成要素に分解します。これらのパースされた名前要素は、名前のマッチング、名前の正規化、複数レコード名の統合など、他の自動化処理に使用できます。

OpenNameParser は、次の処理を行います。

- 名前が担う機能を示すために、その名前のタイプを特定します。名前エンティティタイプは、個人名と企業名の 2 つのグループに分かれます。それぞれのグループには、さらに複数のサブグループがあります。

- パーシングに使う構文を把握するために、名前の形式を特定します。個人名は、通常、自然な (署名) 順序または逆の順序に従います。企業名は、通常、階層型の順序に従います。
- 名前を構成する各要素が名前全体に占める構文上の関連性を識別するために、要素を特定してラベル付けします。個人名の構文は、敬称、名、ミドルネーム、姓、接尾語、アカウントを示す用語、その他の個人名要素で構成されます。企業名の構文は、企業名や接尾語などで構成されます。
- 結合された個人名と企業名をパースし、それらを 1 つのレコードとして残すか、複数のレコードに分割します。Examples of conjoined names include "Mr.and Mrs.John Smith" and "Baltimore Gas & Electric dba Constellation Energy".
- 出力をレコードまたはリストとしてパースします。
- パーシングによる訂正の信頼度を示すパーシング スコアを割り当てます。

### リソース URL

```
http://server:port/soap/OpenNameParser
```

### 例

SOAP リクエストを以下に示します。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:open="http://www.pb.com/spectrum/services/OpenNameParser"
xmlns:spec="http://spectrum.pb.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <open:OpenNameParserRequest>
      <open:input_port>
        <open:Input>
          <open:Name>John Williams Smith</open:Name>
        </open:Input>
      </open:input_port>
    </open:OpenNameParserRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

応答は次のようになります。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:OpenNameParserResponse xmlns:ns2="http://spectrum.pb.com/"
xmlns:ns3="http://www.pb.com/spectrum/services/OpenNameParser">
      <ns3:output_port>
        <ns3:Result>
          <ns3:Name>John Williams Smith</ns3:Name>
          <ns3:CultureCodeUsedToParse/>
          <ns3:FirstName>John</ns3:FirstName>
        </ns3:Result>
      </ns3:output_port>
    </ns3:OpenNameParserResponse>
  </soap:Body>
</soap:Envelope>
```



```

      <ns3:LastName>Smith</ns3:LastName>
      <ns3:MiddleName>Williams</ns3:MiddleName>
      <ns3:Names/>
      <ns3:IsParsed>true</ns3:IsParsed>
      <ns3:IsPersonal>true</ns3:IsPersonal>
      <ns3:IsConjoined>false</ns3:IsConjoined>
      <ns3:IsReverseOrder>false</ns3:IsReverseOrder>
      <ns3:IsFirm>false</ns3:IsFirm>
      <ns3:NameScore>100</ns3:NameScore>
      <ns3:user_fields/>
    </ns3:Result>
  </ns3:output_port>
</ns3:OpenNameParserResponse>
</soap:Body>
</soap:Envelope>

```

## 要求

入力データのパラメータ

表 154 : Open Name Parser の入力

パラメータ	説明
CultureCode	<p>入力された名前データのカルチャー。オプションは次のとおりです。</p> <p><b>Null (empty)</b>      グローバル カルチャー (デフォルト)。</p> <p><b>de</b>                      ドイツ語。</p> <p><b>es</b>                      スペイン語。</p> <p><b>ja</b>                      日本語。</p> <p>注 : Open Parser ドメインエディタを使用して独自のドメインを追加した場合、そのドメインのカルチャーとカルチャー コードも有効になります。</p>
Name	<p>パースしたい名前。このフィールドは必須です。</p>

## オプション

パーシング オプションのパラメータ

次の表に、名前のパーシングを制御するオプションを示します。



表 155 : Open Name Parser パーシング オプション

パラメータ	説明
ParseNaturalOrderPersonalNames	<p>敬称、名、ミドル ネーム、姓、および接尾語の順序で名前をパースするかどうかを指定します。</p> <p><b>true</b>      正順序の個人名をパースします。</p> <p><b>false</b>     正順序の名前をパースしません。</p>
ParseReverseOrderPersonalNames	<p>姓が最初に指定されている名前をパースするかどうかを指定します。</p> <p><b>true</b>      逆順序の個人名をパースします。</p> <p><b>false</b>     逆順序の名前をパースしません。</p>
ParseConjoinedNames	<p>結合名をパースするかどうかを指定します。</p> <p><b>true</b>      結合名をパースします。</p> <p><b>false</b>     結合名をパースしません。</p>
SplitConjoinedNames	<p>Bill &amp; Sally Smith など、複数の人物を含む結合名を複数のレコードに分割するかどうかを指定します。</p> <p><b>true</b>      結合名を分割します。</p> <p><b>false</b>     結合名を分割しません。</p>
ParseBusinessNames	<p>企業名をパースするかどうかを指定します。</p> <p><b>true</b>      企業名をパースします。</p> <p><b>false</b>     企業名をパースしません。</p>
OutputAsList	<p>パース済み名前要素をリスト形式で返すかどうかを指定します。</p> <p><b>true</b>      パース済み要素をリスト形式で返します。</p> <p><b>false</b>     パース済み要素をリスト形式で返しません。</p>

パラメータ	説明
ShortcutThreshold	パフォーマンスと品質のバランスをとる方法を指定します。パフォーマンスを上げると、品質出力が下がります。同様に、品質を上げると、パフォーマンスが下がります。このしきい値を満たすと、レコードに対して他の処理は実行されません。 0 ~ 100 の値を指定します。デフォルト値は 100 です。

### カルチャー オプションのパラメータ

次の表に、名前カルチャーを制御するオプションを示します。

**表 156 : Open Name Parser カルチャー オプション**

パラメータ	説明
DefaultCulture	パーシング グラマーに含めるカルチャーを指定します。デフォルトでは、グローバルカルチャーが選択されます。 カンマ区切りリストに 2 文字のカルチャー コードを優先する順に指定することで、カルチャーを指定します。例えば、まずはスペインのカルチャー、次に日本のカルチャーを使用して名前のパースを試みるには、次のように指定します。 <code>es, ja, ,</code>

### 詳細オプションのパラメータ

次の表に、名前パーシング用の詳細オプションを示します。

**表 157 : Open Name Parser の詳細オプション**

オプション	説明
NaturalOrderPersonalNamesDomain	正順序個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。

## オプション

## 説明

---

**NaturalOrderPersonalNamesPriority**

使用する他のドメインに対する正順序個人名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。

ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。

---

**ReverseOrderPersonalNamesDomain**

逆順序個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。

---

**ReverseOrderPersonalNamesPriority**

使用する他のドメインに対する逆順序個人名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。

ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。

---

**NaturalOrderConjoinedPersonalNamesDomain**

正順序結合個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。

---

## オプション

## 説明

---

**NaturalOrderConjoinedPersonalNamesPriority**

使用する他のドメインに対する正順序結合個人名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。

ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。

---

**ReverseOrderConjoinedPersonalNamesDomain**

逆順序結合個人名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。

---

**ReverseOrderConjoinedPersonalNamesPriority**

使用する他のドメインに対する逆順序結合個人名ドメインの優先度を示す1～5の数値を指定します。これにより、実行するパーサーの順序が決定されます。

ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。

---

**BusinessNamesDomain**

企業名のパース時に使用するドメインを指定します。有効な値は、Enterprise Designer の Open Parser ドメイン エディタ ツールで定義されたドメイン名です。

---

## オプション

## 説明

## BusinessNamesPriority

使用する他のドメインに対する企業名ドメインの優先度を示す 1～5 の数値を指定します。これにより、実行するパーサーの順序が決定されます。

ショートカットしきい値のオプションに設定された数字よりもスコアの高い最初のドメインに対して結果が返されます。そのしきい値に達しているドメインがない場合は、スコアの最も高いドメインに対する結果が返されます。複数のドメインが同時にしきい値に達している場合は、最初に実行された(ここで設定された順序によって決まる)ドメインが優先され、その結果が返されます。

## 応答

表 158 : Open Name Parser の出力

応答要素	書式	説明
AccountDescription	文字列	名前の一部であるアカウント説明。例えば、"Mary Jones Account # 12345" で、アカウント説明は "Account#12345"。
Names	文字列	パース済み要素のリストを含む階層フィールド。このフィールドは、[パーシングオプション]の[結果をリストに出力]ボックスをチェックしている場合に返されます。
会社名関係のフィールド		
FirmConjunction	文字列	"d/b/a" (doing business as)、"o/a" (operating as)、"t/a" (trading as) などの略語を含む企業の名前を示します。
FirmName	文字列	会社名。例えば、"Pitney Bowes"。
FirmSuffix	文字列	会社名の接尾語。例えば、"Co."、"Inc."

応答要素	書式	説明
IsFirm	文字列	名前が、個人名ではなく、企業名であることを示します。
<b>個人名に関するフィールド</b>		
Conjunction	文字列	名前に、"and"、"or"、"&" などの接続詞が含まれることを示します。
CultureCode	文字列	入力データに含まれるカルチャー コード。
CultureCodeUsedToParse	文字列	データのパーズに使用されたカルチャー固有のグラマーを特定します。 <b>Null (empty)</b> グローバル カルチャー (デフォルト)。 <b>de</b> ドイツ語。 <b>es</b> スペイン語。 <b>ja</b> 日本語。  <b>注：</b> Open Parser ドメインエディタを使用して独自のドメインを追加した場合、そのドメインのカルチャーとカルチャーコードもこのフィールドに表示されます。
FirstName	文字列	個人のファースト ネーム。
GeneralSuffix	文字列	個人名の一般/職業接尾語。例えば、 MD PhD 。
IsParsed	文字列	出力レコードがパーズされたかどうかを示します。値は <b>True</b> または <b>False</b> です。
IsPersonal	文字列	名前が企業名ではなく、個人名であるかどうかを示します。値は <b>True</b> または <b>False</b> です。
IsReverseOrder	文字列	入力名が逆順序であるかどうかを示します。値は <b>True</b> または <b>False</b> です。

応答要素	書式	説明
LastName	文字列	個人名のラスト ネーム。父方の姓が含まれます。
LeadingData	文字列	名前の前に付けられる、名前以外の情報。
MaturitySuffix	文字列	個人の世代/家族接尾語。例えば、Jr.または Sr.。
MiddleName	文字列	個人のみドル ネーム。
Name.	文字列	入力に指定された個人名または企業名。
NameScore	文字列	各名前の既知および不明トークンの平均スコアを示します。NameScore の値は、パーシング グラマーでの定義に従って、0 ~ 100 の間になります。マッチが返されない場合は、0 が返されます。
SecondaryLastName	文字列	スペイン語のパーシング グラマーでは、その人の母の姓。
TitleOfRespect	文字列	"Mr."、"Mrs."、"Dr." など、名前の前に付けられる情報。
TrailingData	文字列	名前の後に付けられる、名前以外の情報。
結合名関係のフィールド		
Conjunction2	文字列	結合されている 2 番目の名前に、"and"、"or"、"&" などの接続詞が含まれることを示します。
Conjunction3	文字列	結合されている 3 番目の名前に、"and"、"or"、"&" などの接続詞が含まれることを示します。

応答要素	書式	説明
FirmName2	文字列	結合されている 2 番目の企業名。例えば、Baltimore Gas & Electric dba Constellation Energy。
FirmSuffix2	文字列	結合されている 2 番目の企業の接尾語。
FirstName2	文字列	結合されている FirstName の 2 番目の名
FirstName3	文字列	結合されている FirstName の 3 番目の名。
GeneralSuffix2	文字列	結合されている 2 番目の名前の一般/職業接尾語。例えば、MD PhD。
GeneralSuffix3	文字列	結合されている 3 番目の名前の一般/職業接尾語。例えば、MD PhD。
IsConjoined	文字列	入力名が結合名であることを示します。結合名は、例えば、"John and Jane Smith"。
LastName2	文字列	結合されている LastName の 2 番目の姓。
LastName3	文字列	結合されている LastName の 3 番目の姓。
MaturitySuffix2	文字列	結合されている 2 番目の名前の世代/家族接尾語。例えば、Jr.または Sr。
MaturitySuffix3	文字列	結合されている 3 番目の名前の世代/家族接尾語。例えば、Jr.または Sr。
MiddleName2	文字列	結合されている MiddleName の 2 番目の名。



応答要素	書式	説明
MiddleName3	文字列	結合されている MiddleName の 3 番目の名。
TitleOfRespect2	文字列	"Mr."、"Mrs."、"Dr." など、結合されている 2 番目の名前の前に付けられる情報。
TitleOfRespect3	文字列	"Mr."、"Mrs."、"Dr." など、結合されている 3 番目の名前の前に付けられる情報。

# 付録

## このセクションの構成

---

バッファリング	731
国コード	734
ValidateAddress 確信アルゴリズム	761

# A - バッファリング

## このセクションの構成

---

バッファリング

732

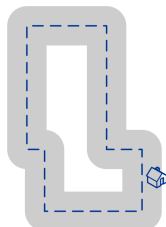
## バッファリング

ポリゴン、ライン、ポイントに近いエリアを指定するのにバッファリングを使用します。

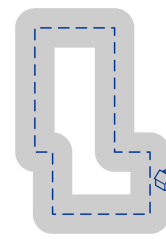
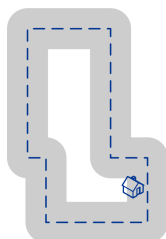
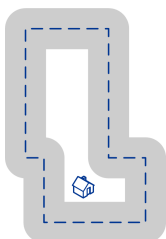


バッファリングされたポリゴン (zone) バッファリングされたライン (corridor) バッファリングされたポイント (circle)

例えば、あなたが保険会社に勤務している場合、潜在的な顧客の家が洪水発生地帯の 500 フィート以内にあるかどうかを知ることができれば、実際に洪水発生地帯内になかったとしても、洪水保険を勧めることができます。下記の図は、バッファリングされたポリゴンを使用してこのシナリオを表しています。点線は、洪水発生地帯の境界線を示しており、色の付いているエリアは境界線から 500 フィートの範囲にあるエリアを示しています。



バッファエリアは、境界線の両サイド (内側と外側) に及んでいます。バッファリングを使用する場合、出力フィールド **BufferRelation** はそのポイントがバッファリングに含まれているかどうか、また、ポリゴンの内側と外側のどちらにあるかを示します。以下の図は、これを表したものです。



ポイントはポリゴンの中にあり、バッファエリア内にはありません。	ポイントは、ポリゴンの中にあり、バッファエリア内にあります。	ポイントは、ポリゴンの外にあります。バッファエリア内にあります。
出力フィールド <code>BufferRelation</code> は "P" となります。	出力フィールド <code>BufferRelation</code> は "I" となります。	出力フィールド <code>BufferRelation</code> は "B" となります。

---

レコードごとのポリゴン バッファのサイズは `BufferWidth` 入力フィールドで指定し、ジョブ用のデフォルトのポリゴン バッファ幅は `DefaultBufferWidth`[デフォルト バッファ幅] オプションで設定します。

# B - 国コード

## このセクションの構成

---

ISO 国コードとモジュール サポート

735

## ISO 国コードとモジュール サポート

この表に、各国の ISO コードと、各国の住所作成、ジオコーディング、およびルーティングをサポートするモジュールを示します。

Enterprise Geocoding モジュールにアフリカ (30 か国)、中東 (8 か国)、ラテンアメリカ (20 か国) のデータベースが含まれていることに注意してください。これらのデータベースは、国別のジオコーディングデータベースがない、各地域の比較的小さな国をカバーします。[サポートされるモジュール] 列は、これらのアフリカ、中東、ラテンアメリカ データベースに含まれる国を示しています。

また、Geocode Address World データベースは、すべての国について地図上の限定的な郵便ジオコーディング (通り レベルではない) を提供します。

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Afghanistan	AF	AFG	Universal Addressing モジュール
Aland Islands	AX	ALA	Universal Addressing モジュール
アルバニア	AL または SQ (Routing)	ALB	Universal Addressing モジュール Enterprise Geocoding モジュール Enterprise Routing モジュール
アルジェリア	DZ	DZA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
American Samoa	AS	ASM	Universal Addressing モジュール
Andorra	AD	AND	Enterprise Geocoding モジュール(アンドラは、スペインのジオコードで扱われています)。 Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Angola	AO	AGO	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Anguilla	AI	AIA	Universal Addressing モジュール
Antarctica	AQ	ATA	Universal Addressing モジュール
Antigua And Barbuda	AG	ATG	Universal Addressing モジュール
アルゼンチン	AR	ARG	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール
Armenia	AM	ARM	Universal Addressing モジュール
Aruba	AW	ABW	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
オーストラリア	AU	AUS	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
オーストリア	AT	AUT	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Azerbaijan	AZ	AZE	Universal Addressing モジュール



ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
バハマ	BS	BHS	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール
Bahrain	BH	BHR	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
Bangladesh	BD	BGD	Universal Addressing モジュール
Barbados	BB	BRB	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Belarus	BY	BLR	Universal Addressing モジュール Enterprise Routing モジュール
Belgium	BE	BEL	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Belize	BZ	BLZ	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Benin	BJ	BEN	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Bermuda	BM	BMU	Universal Addressing モジュール Enterprise Routing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Bhutan	BT	BTN	Universal Addressing モジュール
ボリビア	BO	BOL	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Bonaire, Saint Eustatius And Saba	BQ	BES	Universal Addressing モジュール
Bosnia And Herzegovina	BA	BIH	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール Enterprise Geocoding モジュール
Botswana	BW	BWA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Bouvet Island	BV	BVT	Universal Addressing モジュール
ブラジル	BR	BRA	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
British Indian Ocean Territory	IO	IOT	Universal Addressing モジュール
Brunei Darussalam	BN	BRN	Enterprise Geocoding モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Bulgaria	BG	BGR	Enterprise Geocoding モジュール Universal Addressing モジュール
Burkina Faso	BF	BFA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Burundi	BI	BDI	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Cambodia	KH	KHM	Universal Addressing モジュール
Cameroon	CM	CMR	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
カナダ	CA	CAN	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Cape Verde	CV	CPV	Universal Addressing モジュール
Cayman Islands	KY	CYM	Universal Addressing モジュール
Central African Republic	CF	CAF	Universal Addressing モジュール
Chad	TD	TCD	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
チリ	CL	CHL	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール
中国	CN または zh_CN (Routing)	CHN	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール
Christmas Island	CX	CXR	Universal Addressing モジュール
Cocos (Keeling) Islands	CC	CCK	Universal Addressing モジュール
コロンビア	CO	COL	Enterprise Geocoding モジュール Universal Addressing モジュール
Comoros	KM	COM	Universal Addressing モジュール
Congo, Republic Of The	CG	COG	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Congo, The Democratic Republic Of The	CD	COD	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール Enterprise Routing モジュール
Cook Islands	CK	COK	Universal Addressing モジュール
Costa Rica	CR	CRI	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Côte d'Ivoire	CI	CIV	Universal Addressing モジュール
Croatia	HR	HRV	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Cuba	CU	CUB	Enterprise Geocoding モジュール (ラテンアメリカ) Enterprise Routing モジュール Universal Addressing モジュール
Curacao	CW	CUW	Universal Addressing モジュール
キプロス	CY	CYP	Enterprise Geocoding モジュール Universal Addressing モジュール
チェコ共和国	CZ または CS (Routing)	CZE	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール
デンマーク	DK	DNK	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Djibouti	DJ	DJI	Universal Addressing モジュール
Dominica	DM	DMA	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Dominican Republic	DO	DOM	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Ecuador	EC	ECU	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Egypt	EG	EGY	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
El Salvador	SV	SLV	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Equatorial Guinea	GQ	GNQ	Universal Addressing モジュール
Eritrea	ER	ERI	Universal Addressing モジュール
Estonia	EE	EST	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Ethiopia	ET	ETH	Universal Addressing モジュール
Falkland Islands (Malvinas)	FK	FLK	Universal Addressing モジュール
Faroe Islands	FO	FRO	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Fiji	FJ	FJI	Universal Addressing モジュール
フィンランド	FI	FIN	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
フランス	FR	FRA	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
French Guiana	GF	GUF	Enterprise Geocoding モジュール (フランス領 ギアナは、フランスのジオコードで扱われて います)。 Universal Addressing モジュール
French Polynesia	PF	PYF	Universal Addressing モジュール
French Southern Territories	TF	ATF	Universal Addressing モジュール
Gabon	GA	GAB	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Gambia	GM	GMB	Universal Addressing モジュール
Georgia	GE	GEO	Universal Addressing モジュール
ドイツ	DE	DEU	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Ghana	GH	GHA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール Enterprise Routing モジュール
Gibraltar	GI	GIB	Enterprise Geocoding モジュール (ジブラルタルは、スペインのジオコードで扱われています)。 Universal Addressing モジュール
Greece	GR	GRC	Enterprise Geocoding モジュール Universal Addressing モジュール
Greenland	GL	GRL	Universal Addressing モジュール
Grenada	GD	GRD	Universal Addressing モジュール
Guadeloupe	GP	GLP	Enterprise Geocoding モジュール (グアドループは、フランスのジオコードで扱われています)。 Universal Addressing モジュール
Guam	GU	GUM	Universal Addressing モジュール
Guatemala	GT	GTM	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Guernsey	GG	GGY	Universal Addressing モジュール



ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Guinea	GN	GIN	Universal Addressing モジュール
Guinea-Bissau	GW	GNB	Universal Addressing モジュール
Guyana	GY	GUY	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Haiti	HT	HTI	Universal Addressing モジュール
Heard Island and McDonald Islands	HM	HMD	Universal Addressing モジュール
Holy See (Vatican City State)	VA	付加価値税	Enterprise Geocoding モジュール (バチカン は、イタリアのジオコードで扱われています)。 Universal Addressing モジュール
Honduras	HN	HND	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Hong Kong	HK	HKG	Enterprise Geocoding モジュール Universal Addressing モジュール
ハンガリー	HU	HUN	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
アイスランド	IS	ISL	Enterprise Geocoding モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
インド	IN	IND	Enterprise Geocoding モジュール Universal Addressing モジュール
Indonesia	ID	IDN	Enterprise Geocoding モジュール Universal Addressing モジュール
Iran, Islamic Republic Of	IR	IRN	Universal Addressing モジュール
イラク	IQ	IRQ	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
アイルランド	IE	IRL	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Isle Of Man	IM	IMN	Universal Addressing モジュール
Israel	IL	ISR	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール
Italy	IT	ITA	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Jamaica	JM	JAM	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
日本	JP	JPN	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Jersey	JE	JEY	Universal Addressing モジュール
ヨルダン	JO	JOR	Universal Addressing モジュール Enterprise Geocoding モジュール (中東) Enterprise Routing モジュール
Kazakhstan	KZ	KAZ	Universal Addressing モジュール
Kenya	KE	KEN	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール Enterprise Routing モジュール
Kiribati	KI	KIR	Universal Addressing モジュール
Korea, Democratic People's Republic Of	KP	PRK	Universal Addressing モジュール
Korea, Republic Of	KR	KOR	Enterprise Geocoding モジュール Universal Addressing モジュール
コソボ	Xk	XKX	Enterprise Geocoding モジュール Universal Addressing モジュール
Kuwait	KW	KWT	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Kyrgyzstan	KG	KGZ	Universal Addressing モジュール
Lao People's Democratic Republic	LA	LAO	Universal Addressing モジュール
Latvia	LV	LVA	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Lebanon	LB	LBN	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
Lesotho	LS	LSO	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール Enterprise Routing モジュール
Liberia	LR	LBR	Universal Addressing モジュール
Libyan Arab Jamahiriya	LY	LBY	Universal Addressing モジュール
Liechtenstein	LI	LIE	Enterprise Geocoding モジュール (リヒテンシュタインは、スイスのジオコードで扱われています)。 Enterprise Routing モジュール Universal Addressing モジュール
Lithuania	LT	LTU	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Luxembourg	LU	LUX	Enterprise Geocoding モジュール (ルクセンブルクは、ベルギーのジオコードで扱われています)。 Enterprise Routing モジュール Universal Addressing モジュール
Macao	MO	MAC	Enterprise Geocoding モジュール Universal Addressing モジュール
Macedonia, Former Yugoslav Republic Of	MK	MKD	Enterprise Geocoding モジュール Universal Addressing モジュール
Madagascar	MG	MDG	Universal Addressing モジュール
Malawi	MW	MWI	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
マレーシア	MY	MYS	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Maldives	MV	MDV	Universal Addressing モジュール
Mali	ML	MLI	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
マルタ	ML	MLT	Enterprise Geocoding モジュール Universal Addressing モジュール
Marshall Islands	MH	MHL	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Martinique	MQ	MTQ	Enterprise Geocoding モジュール (マルティニークは、フランスのジオコードで扱われています)。 Universal Addressing モジュール
Mauritania	MR	MRT	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Mauritius	MU	MUS	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Mayotte	YT	MYT	Enterprise Geocoding モジュール (マヨットは、フランスのジオコードで扱われています)。 Universal Addressing モジュール
メキシコ	MX	MEX	Enterprise Geocoding モジュール Universal Addressing モジュール
Micronesia, Federated States Of	FM	FSM	Universal Addressing モジュール
Moldova, Republic Of	MD	MDA	Universal Addressing モジュール Enterprise Routing モジュール
Monaco	MC	MCO	Enterprise Geocoding モジュール (モナコはフランスのジオコードで扱われています)。 Universal Addressing モジュール
Mongolia	MN	MNG	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
モンテネグロ	ME	MNE	Enterprise Geocoding モジュール Universal Addressing モジュール
Montserrat	MS	MSR	Universal Addressing モジュール
Morocco	MA	MAR	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Mozambique	MZ	MOZ	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール Enterprise Routing モジュール
Myanmar	MM	MMR	Universal Addressing モジュール
Namibia	NA	NAM	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Nauru	NR	NRU	Universal Addressing モジュール
Nepal	NP	NPL	Universal Addressing モジュール
Netherlands	NL	NLD	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
New Caledonia	NC	NCL	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
ニュージーランド	NZ	NZL	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Nicaragua	NI	NIC	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Niger	NE	NER	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Nigeria	NG	NGA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Niue	NU	NIU	Universal Addressing モジュール
Norfolk Island	NF	NFK	Universal Addressing モジュール
Northern Mariana Islands	MP	MNP	Universal Addressing モジュール
ノルウェー	NO	NOR	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Oman	OM	OMN	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
Pakistan	PK	PAK	Universal Addressing モジュール



ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Palau	PW	PLW	Universal Addressing モジュール
Palestinian Territory, Occupied	PS	PSE	Universal Addressing モジュール
Panama	PA	PAN	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Papua New Guinea	PG	PNG	Universal Addressing モジュール
Paraguay	PY	PRY	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Peru	PE	PER	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Philippines	PH	PHL	Enterprise Geocoding モジュール Universal Addressing モジュール Enterprise Routing モジュール
Pitcairn	PN	PCN	Universal Addressing モジュール
ポーランド	PL	POL	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
ポルトガル	PT	PRT	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Puerto Rico	PR	PRI	Universal Addressing モジュール
Qatar	QA	QAT	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
Reunion	RE	REU	Enterprise Geocoding モジュール (レユニオンはフランスのジオコードで扱われています)。 Universal Addressing モジュール
ルーマニア	RO	ROU	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Russian Federation	RU	RUS	Enterprise Geocoding モジュール Universal Addressing モジュール
Rwanda	RW	RWA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Saint Barthelemy	BL	BLM	Universal Addressing モジュール
Saint Helena, Ascension and Tristan Da Cunha	SH	SHE	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Saint Kitts and Nevis	KN	KNA	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Saint Lucia	LC	LCA	Universal Addressing モジュール
Saint Martin (French Part)	MF	MAF	Universal Addressing モジュール
Saint Pierre and Miquelon	PM	SPM	Universal Addressing モジュール
Saint Vincent and the Grenadines	VC	VCT	Universal Addressing モジュール
Samoa	WS	WSM	Universal Addressing モジュール
San Marino	SM	SMR	Enterprise Geocoding モジュール (サンマリノは、イタリアのジオコードで扱われています)。 Universal Addressing モジュール
Sao Tome and Principe	ST	STP	Universal Addressing モジュール
Saudi Arabia	SA	SAU	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
Senegal	SN	SEN	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
セルビア	RS	SRB	Enterprise Geocoding モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Seychelles	SC	SYC	Universal Addressing モジュール
Sierra Leone	SL	SLE	Universal Addressing モジュール
シンガポール	SG	SGP	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Sint Maarten (Dutch Part)	SX	SXM	Universal Addressing モジュール
Slovakia	SK	SVK	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Slovenia	SI	SVN	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Solomon Islands	SB	SLB	Universal Addressing モジュール
Somalia	SO	SOM	Universal Addressing モジュール
South Africa	ZA	ZAF	Enterprise Geocoding モジュール Universal Addressing モジュール
South Georgia And The South Sandwich Islands	GS	SGS	Enterprise Geocoding モジュール Universal Addressing モジュール
South Sudan	SS	SSD	Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Spain	ES	ESP	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Sri Lanka	LK	LKA	Universal Addressing モジュール
Sudan	SD	SDN	Universal Addressing モジュール
Suriname	SR	SUR	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Svalbard And Jan Mayen	SJ	SJM	Universal Addressing モジュール
Swaziland	SZ	SWZ	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
スウェーデン	SE	SWE	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Switzerland	CH	CHE	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Syrian Arab Republic	SY	SYR	Universal Addressing モジュール
Taiwan, Province of China	TW または zh_TW (Routing)	TWN	Universal Addressing モジュール Enterprise Routing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Tajikistan	TJ	TJK	Universal Addressing モジュール
Tanzania, United Republic Of	TZ	TZA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール Enterprise Routing モジュール
Thailand	TH	THA	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
Timor-Leste	TL	TLS	Universal Addressing モジュール
Togo	TG	TGO	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Tokelau	TK	TKL	Universal Addressing モジュール
Tonga	TO	TON	Universal Addressing モジュール
Trinidad and Tobago	TT	TTO	Enterprise Geocoding モジュール (ラテンアメリカ) Universal Addressing モジュール
Tunisia	TN	TUN	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Turkey	TR	TUR	Enterprise Geocoding モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Turkmenistan	TM	TKM	Universal Addressing モジュール
Turks And Caicos Islands	TC	TCA	Universal Addressing モジュール
Tuvalu	TV	TUV	Universal Addressing モジュール
Uganda	UG	UGA	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Ukraine	UA	UKR	Enterprise Geocoding モジュール Universal Addressing モジュール
United Arab Emirates	AE	ARE	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
英国	GB	GBR	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
米国	US	USA	Enterprise Geocoding モジュール Enterprise Routing モジュール Universal Addressing モジュール
United States Minor Outlying Islands	UM	UMI	Universal Addressing モジュール
ウルグアイ	UY	URY	Enterprise Geocoding モジュール Universal Addressing モジュール

ISO 国名	ISO 3166-1 Alpha-2	ISO 3166-1 Alpha-3	サポートされるモジュール
Uzbekistan	UZ	UZB	Universal Addressing モジュール
Vanuatu	VU	VUT	Universal Addressing モジュール
Venezuela, Bolivarian Republic Of	VE	VEN	Enterprise Geocoding モジュール Universal Addressing モジュール
Viet Nam	VN	VNM	Enterprise Geocoding モジュール Universal Addressing モジュール
Virgin Islands, British	VG	VGB	Universal Addressing モジュール
Virgin Islands, U.S.	VI	VIR	Universal Addressing モジュール
Wallis and Futuna	WF	WLF	Universal Addressing モジュール
Western Sahara	EH	ESH	Universal Addressing モジュール
イエメン	YE	YEM	Enterprise Geocoding モジュール (中東) Universal Addressing モジュール
Zambia	ZM	ZMB	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール
Zimbabwe	ZW	ZWE	Enterprise Geocoding モジュール (アフリカ) Universal Addressing モジュール



# C - ValidateAddress 確信アルゴリズム

## このセクションの構成

---

Validate Address 確信アルゴリズムの概要	762
米国とカナダの住所に対する確信アルゴリズム住所の検証	762
国際住所用の確信アルゴリズム	763

## Validate Address 確信アルゴリズムの概要

ValidateAddress は、妥当性を確認した各住所に対し、確信スコアを計算します。このスコアは、妥当性を確認した住所がどの程度正しいかを表します。確信コードの値は 0 ~ 100 で、0 はまったく確信がなく、100 はマッチ結果が正しいことに対する確信レベルが非常に高いことを表します。確信コードは、個々の出力フィールドに対するマッチ結果を考慮に入れたアルゴリズムに基づいて計算されます。この計算に関連する出力フィールドは、以下のとおりです。

- Country
- City
- State
- PostalCode
- StreetName
- HouseNumber
- LeadingDirectional
- TrailingDirectional
- StreetSuffix
- ApartmentNumber

アルゴリズムにおいて、各フィールドにはそれぞれ独自の重みがあります。また、各フィールドに対し、"Success"、"Failure"、または "Changed" というマッチ結果が存在します("Changed" は、マッチさせるためにフィールドの内容を修正した場合に該当します)。マッチ結果 ("Success"、"Failure"、または "Changed") によって、そのフィールドの係数が決まります。以上より、確信コードは、重みと係数の積を用いて、次のように算出します。

```
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
```

## 米国とカナダの住所に対する確信アルゴリズム住所の検証

以下の表で、米国およびカナダの住所に対する ValidateAddress の確信アルゴリズムのスコアリングとロジックについて説明します。

表 159 : 米国とカナダの住所に対する確信アルゴリズム

フィールド	重み付け/マッチ スコア	変更の場合の係数 <sup>1</sup>	埋めた場合の係数 <sup>2</sup>
Country	10	100%	0%
City	10	50%	75%
州	15	50%	75%
PostalCode	15	25%	25%
StreetName	15	50%	75%
HouseNumber	15	50%	75%
Directionals	10	50%	75%
StreetSuffix	5	50%	75%
ApartmentNumber	5	50%	75%

## 国際住所用の確信アルゴリズム

米国およびカナダ以外の住所用に 2 つの確信アルゴリズムがあります。1 つは郵便番号を使用する国の住所用で、もう 1 つは郵便番号を使用しない国の住所用です。

<sup>2</sup> このフィールドに入力データが存在しないが、マッチを得るためにデータが埋められた場合のこと。

<sup>1</sup> このフィールドの入力データが、マッチを得るために変更された場合のこと。

以下の表に、郵便番号を使用している国の米国およびカナダ以外の住所用の確信アルゴリズムの詳細を示します。

表 160 : 郵便番号がある国用の確信アルゴリズム

フィールド	重み付け/マッチ スコア	変更の場合の係数 <sup>3</sup>	埋めた場合の係数 <sup>4</sup>	郵便データが使用できない場合の係数
Country	11.11111111111111	100%	0%	0%
City	11.11111111111111	50%	75% <sup>5</sup>	0%
州	16.66666666666667	100%	100	80%
PostalCode	16.66666666666667	100%	100%	80%
StreetName	16.66666666666667	50%	75%	50%
HouseNumber	16.66666666666667	50%	75%	50%
Directionals	0	50%	75%	0%
StreetSuffix	5.55555555555556	50%	75%	50%

<sup>4</sup> このフィールドに入力データが存在しないが、マッチを得るためにデータが埋められた場合のこと。

<sup>3</sup> このフィールドの入力データが、マッチを得るために変更された場合のこと。

<sup>5</sup> カテゴリ C の国の場合、この値は 50% です。各国は、次のいずれかのカテゴリに分類されます。

- **カテゴリ A** — 住所の郵便番号、都市名、州/郡名、通りの住所要素、および国名の検証と修正が可能です。
- **カテゴリ B** — 住所の郵便番号、都市名、州/郡名、および国名の検証と修正が可能です。通りの住所要素の検証または修正はサポートしません。
- **カテゴリ C** — 国名の検証および修正と、郵便番号の書式の検証が可能です。

フィールド	重み付け/マッチ スコア	変更の場合の係数 <sup>3</sup>	埋めた場合の係数 <sup>4</sup>	郵便データが使用できない場合の係数
ApartmentNumber	5.55555555555556	50%	75%	50%

郵便番号を使用しない国用の確信アルゴリズムの詳細を次の表に示します。

表 161 : 郵便番号がない国用の確信アルゴリズム

フィールド	重み付け/マッチ スコア	変更の場合の係数 <sup>6</sup>	埋めた場合の係数 <sup>7</sup>	郵便データが使用できない場合の係数
Country	13.33333333333333	100%	0%	0%
City	13.33333333333333	50%	75% <sup>8</sup>	0%
州	20	100%	100	80%
StreetName	20	50%	75%	50%

<sup>4</sup> このフィールドに入力データが存在しないが、マッチを得るためにデータが埋められた場合のこと。

<sup>3</sup> このフィールドの入力データが、マッチを得るために変更された場合のこと。

<sup>7</sup> このフィールドに入力データが存在しないが、マッチを得るためにデータが埋められた場合のこと。

<sup>6</sup> このフィールドの入力データが、マッチを得るために変更された場合のこと。

<sup>8</sup> カテゴリ C の国の場合、この値は 50% です。各国は、次のいずれかのカテゴリに分類されます。

- **カテゴリ A**—住所の郵便番号、都市名、州/郡名、通りの住所要素、および国名の検証と修正が可能です。
- **カテゴリ B**—住所の郵便番号、都市名、州/郡名、および国名の検証と修正が可能です。通りの住所要素の検証または修正はサポートしません。
- **カテゴリ C**—国名の検証および修正と、郵便番号の書式の検証が可能です。

フィールド	重み付け/マッチ スコア	変更の場合の係数 <sup>6</sup>	埋めた場合の係数 <sup>7</sup>	郵便データが使用できない場合の係数
HouseNumber	20	50%	75%	50%
Directionals	0	50%	75%	0%
StreetSuffix	6.66666666666667	50%	75%	50%
ApartmentNumber	6.66666666666667	50%	75%	50%

郵便番号がない国の一覧を以下の表に示します。

表 162 : 郵便番号がない国

Afghanistan	アルバニア	Angola
Anguilla	バハマ	Barbados
Belize	Benin	Bhutan
Botswana	Burkina Faso	Burundi
Cameroon	Cayman Islands	Central African Rep.
Chad	Cocos Islands	コロンビア

<sup>7</sup> このフィールドに入力データが存在しないが、マッチを得るためにデータが埋められた場合のこと。

<sup>6</sup> このフィールドの入力データが、マッチを得るために変更された場合のこと。

Comoros	Congo (Dem.Rep.)	Congo (Rep.)
Cote d'Ivoire	Korea (North)	Djibouti
Dominica	Equatorial Guinea	Eritrea
Fiji	Gabon	Gambia
Ghana	Grenada	Guyana
アイルランド	Jamaica	Kiribati
リビア	Malawi	Mali
Mauritania	Namibia	Nauru
Palau	Panama	Peru
Qatar	Rwanda	Saint Lucia
Saint Vincent and the Grenadines	Samoa	Sao Tome and Principe
Seychelles	Sierra Leone	Suriname
Tanzania	Timor	Togo
Tonga	Trinidad & Tobago	Tuvalu
Uganda	United Arab Emirates	Vanuatu

イエメン

Zimbabwe

---



# 著作権に関する通知

© 2019 Pitney Bowes. All rights reserved. MapInfo および Group 1 Software は Pitney Bowes Software Inc. の商標です。その他のマークおよび商標はすべて、それぞれの所有者の資産です。

### USPS® 情報

Pitney Bowes Inc. は、ZIP + 4® データベースを光学および磁気媒体に発行および販売する非独占的ライセンスを所有しています。CASS、CASS 認定、DPV、eLOT、FASTforward、First-Class Mail、Intelligent Mail、LACS<sup>Link</sup>、NCOA<sup>Link</sup>、PAVE、PLANET Code、Postal Service、POSTNET、Post Office、RDI、Suite<sup>Link</sup>、United States Postal Service、Standard Mail、United States Post Office、USPS、ZIP Code、および ZIP + 4 の各商標は United States Postal Service が所有します。United States Postal Service に帰属する商標はこれに限りません。

Pitney Bowes Inc. は、NCOA<sup>Link</sup>® 処理に対する USPS® の非独占的ライセンスを所有しています。

Pitney Bowes Software の製品、オプション、およびサービスの価格は、USPS® または米国政府によって規定、制御、または承認されるものではありません。RDI™ データを利用して郵便送料を判定する場合に、使用する郵便配送業者の選定に関するビジネス上の意思決定が USPS® または米国政府によって行われることはありません。

### データ プロバイダおよび関連情報

このメディアに含まれて、Pitney Bowes Software アプリケーション内で使用されるデータ製品は、各種商標によって、および次の 1 つ以上の著作権によって保護されています。

© Copyright United States Postal Service. All rights reserved.

© 2014 TomTom. All rights reserved. TomTom および TomTom ロゴは TomTom N.V. の登録商標です。

© 2016 HERE

Fuente: INEGI (Instituto Nacional de Estadística y Geografía)

電子データに基づいています。© National Land Survey Sweden.

© Copyright United States Census Bureau

© Copyright Nova Marketing Group, Inc.

このプログラムの一部は著作権で保護されています。© Copyright 1993-2007 by Nova Marketing Group Inc. All Rights Reserved

© Copyright Second Decimal, LLC

© Copyright Canada Post Corporation

この CD-ROM には、Canada Post Corporation が著作権を所有している編集物からのデータが収録されています。

© 2007 Claritas, Inc.

Geocode Address World データ セットには、  
<http://creativecommons.org/licenses/by/3.0/legalcode> に存在するクリエイティブ コモンズ アトリビューション ライセンス (「アトリビューション ライセンス」) の下に提供されている GeoNames Project ([www.geonames.org](http://www.geonames.org)) からライセンス供与されたデータが含まれています。お客様による GeoNames データ (Spectrum™ Technology Platform ユーザ マニュアルに記載) の使用は、アトリビューションライセンスの条件に従う必要があります。お客様と Pitney Bowes Software, Inc. との契約と、アトリビューション ライセンスの間に矛盾が生じる場合は、アトリビューションライセンスのみに基づいてそれを解決する必要があります。お客様による GeoNames データの使用に関しては、アトリビューション ライセンスが適用されるためです。



3001 Summer Street  
Stamford CT 06926-0700  
USA

[www.pitneybowes.com](http://www.pitneybowes.com)