Address Validation Service
Legal and Copyright Notices

Payment

You must remit payment in accordance with the FedEx Service Guide, tariff, service agreement or other terms or instructions provided to you by FedEx from time to time. You may not withhold payment on any shipments because of equipment failure or for the failure of FedEx to repair or replace any equipment.

Inaccurate Invoices

If you generate an inaccurate invoice, FedEx® may bill or refund to you the difference according to the FedEx Service Guide, tariff service agreement or other terms or instructions provided to you by FedEx from time to time. A request for refund on a FedEx shipment must be made in accordance with the applicable Service Guide or terms or instructions provided by FedEx from time to time. A shipment given to FedEx with incorrect information is not eligible for refund under any FedEx money-back guarantee. FedEx may suspend any applicable money-back guarantee in the event of equipment failure or if it becomes inoperative.

Confidential and Proprietary

The information contained in this guide is confidential and proprietary to FedEx Corporate Services, Inc. and its affiliates (collectively “FedEx”). No part of this guide may be distributed or disclosed in any form to any third party without written permission of FedEx. This guide is provided to you and its use is subject to the terms and conditions of the FedEx Automation Agreement. The information in this document may be changed at any time without notice. Any conflict between this guide, the FedEx Automation Agreement and the FedEx Service Guide shall be governed by the FedEx Automation Agreement and the FedEx Service Guide, in that order.

© 2014 - 2015 FedEx. FedEx and the FedEx logo are registered service marks. All rights reserved. Unpublished.

Disclaimer

All Improper Transaction scenarios are for example only. They do not reflect all error condition scenarios.
About This Guide

This guide describes how to integrate with FedEx Web Services. It is written for the application developer who uses web services to design and deploy applications enabled by FedEx. It describes how to get started with application development and how to use the Application Programming Interface (API). It also describes each available service in addition to the business logic that drives each FedEx process.

Document Organization

Each web service provides access to FedEx features. The service description includes service details and a full schema listing to facilitate application development.

Resources

- FedEx Developer Resource Center: fedex.com/developer
- FedEx Services At-a-Glance: fedex.com/us/services
- O’Reilly XML.com: www.xml.com
- Secure Socket Layer Certificates: fedex.com/us/developer/downloads/dev_cert.zip
- Web Services organization home page: www.web-services.org

Support

- Contact FedEx Web Services technical support at webservice@fedex.com.
- Regional contact email addresses are:
  - EMEA - emeaweb.services@fedex.com
  - APAC - apacweb.services@fedex.com
- For technical support, call 1.877.339.2774 and state “API” at the voice prompt.
- Support hours are Monday through Friday, 7:00 a.m. to 9:00 p.m. CST, and Saturday, 9:00 a.m. to 3:00 p.m. CST.
- Customers using a FedEx® Compatible Solutions Program automation solution should contact their software provider for support.
1. Introduction

FedEx Web Services gives you the tools to build custom platform- and interface-independent applications that access FedEx features. You can use FedEx Web Services in a variety of ways to create customized integration solutions for your specific shipping needs. Here are just a few of the ways a company can use web services to streamline operations, improve visibility, and provide more choices to clients:

- **Give Customers More Options:** Help customers learn about all the available shipping options and rates with Ship Service WSDL, OpenShip WSDL, and Rate Services WSDL. You can also extend this service to your shopping cart and website, allowing customers to access money-saving information firsthand.
- **More Convenience:** Use the Locations Service WSDL to find the FedEx pickup location nearest your customer. Or, send an email to your customers with a link to this service as part of your standard order-receipt process.
- **Offer Global Shipping Options:** Create shipping labels for worldwide locations. Improve customer service by offering more shipping options to customers in more countries with the consolidated Ship Service WSDL.
- **Reduce Customer Service Costs:** Decrease phone traffic from customers checking the status of their shipments and cut customer service costs. FedEx provides online Tracking and Visibility Services that allow you to provide customers with the status of shipments, Signature Proof of Delivery (SPOD), and Shipment Notification in the Ship Request.
- **Simplify Processes and Improve Satisfaction:** In addition to ExpressTagAvailability, provide a simple way to allow customers to return an order with Email Labels. This service sends an email with the address (URL) of a website where the recipient can log in and print a return label.

Why should developers be interested in web services?

- **Interoperability:** Any web service can interact with any other web service and can be written in any programming language.
- **Ubiquity:** Web services communicate using HTTP and XML. Any connected device that supports these technologies can both host and access web services.
- **Low Barrier to Entry:** The concepts behind web services are easy to understand, and developers can quickly create and deploy them using many toolkits available on the web.
- **Industry Support:** Major content providers and vendors support the web services movement.

Any application running on any platform can interact with a web service by using the Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) standards for message transfer and service discovery. By following the standards, applications can seamlessly communicate with platform services.

1.1 Document Overview

This guide provides instructions for coding the functions you need to develop FedEx supported applications. The following chapters make up this guide:

- **Introduction (this chapter):**
  - Documentation overview and guidelines, including how to use the Help application and how to print this guide.
  - Overview information about web services, including a high-level description of FedEx Web Services methods.
  - Coding basics.
  - Overview information about testing and certifying your application.

Each chapter covering FedEx Web Services coding includes:

- **Service Details:** Business rules for using the FedEx service.
- **Service Options:** Links to additional services that can be added to the basic web service.
• Coding Details: Best practices information, basic request and reply elements, and a link to error messages.
• XML Schema: A link to the layout for the service. This layout provides coding requirements for all elements in the schema.

1.2 Printing All or Part of This Guide

You can print all or part of this guide from the PDF version.

1.2.1 Printing from the PDF Version

From the PDF version you can print the complete document or a page range of the document.
1) Open the PDF file and click the printer icon or click File > Print.
2) From the Print dialog box, print the complete document, specify a page range, or choose from any of the available print options.

1.3 Web Services, WSDL, and SOAP Overview

This section describes the standard coding technologies used in FedEx Web Services.

1.3.1 Web Services

Web services are a collection of programming technologies, including XML, Web Services Description Language (WSDL), and SOAP, which allow you to build programming solutions for specific messaging and application integration.

Web services are, by definition, platform independent. FedEx Web Services allow developers to build custom applications that are independent of changes to the FedEx interface.

Web Services are consumed by many different applications across many platforms. It is based on the basic principles that govern XML standards, one of which is how Namespaces can be declared and applied. Namespaces are declared as an attribute of an element. It is not mandatory to declare namespaces only at the root element; rather it could be declared at any element in the XML document. The scope of a declared namespace begins at the element where it is declared and applies to the entire content of that element, unless overridden by another namespace declaration with the same prefix name, the content of an element is the content between the <opening-tag> and </closing-tag> of that element. So essentially, XML namespace declarations are scoped, meaning that the declared prefix (or default namespace) is in force for the element on which the declaration occurs (as well as its descendant elements). A namespace declared as follows:

```
<v12:RateReply xmlns:v12="http://
is semantically same as
<RateReply xmlns="http://fedex.com/ws/rate/v12">
or even (hypothetically) same as
<foo:RateReply xmlns:foo="http://fedex.com/ws/rate/v12">
```

1.3.2 WSDL

A SOAP request to, or response from, a service is generated according to the service’s WSDL definition. A WSDL document describes a service. It is an XML document that provides information about what the service does, the methods that are available, their parameters, and parameter types. It describes how to communicate with the service in order to generate a request to, or decipher a response from, the service.

The purpose of a WSDL is to completely describe a web service to a client. A WSDL defines where the service is available and what communications protocol is used to talk to the service. It defines everything required to write a program to work with an XML web service. A WSDL document describes a web service using seven major elements. Elements can be abstract or concrete.

Abstract XML elements describe the web service: <types>, <message>, <operation>, <portType>. Concrete XML elements provide connection details: <service>, <port>, <binding>.

```
### 1.3.2.1 WSDL Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;definitions&gt;</td>
<td>The root element contains name space definitions.</td>
</tr>
<tr>
<td>&lt;portType&gt;</td>
<td>The most important WSDL element. It is a set of all operations that a web service can accept and is a container for &lt;operation&gt; elements. This WSDL element describes a web service, the operations that can be performed, and the messages that are involved, and can be compared to a function library (or a module or a class) in a traditional programming language.</td>
</tr>
<tr>
<td>&lt;types&gt;</td>
<td>Defines variable types used in the web service (both the parameters passed to a function and the type of the value passed back via the response). The data types are described by XML schema. This element contains user-defined data types (in the form of XML schema). For maximum platform neutrality, WSDL uses XML schema syntax to define data types.</td>
</tr>
<tr>
<td>&lt;message&gt;</td>
<td>Defines the data elements of an operation. Each message can consist of one or more parts that can be compared to the parameters of a function call in a traditional programming language.</td>
</tr>
<tr>
<td>&lt;operation&gt;</td>
<td>Child of the &lt;binding&gt; element that defines each operation that the port exposes. This element allows only three messages: Message - Definition Input Message - Data web services receive Output Message - Data web services send Fault Message - Error messages from web services</td>
</tr>
<tr>
<td>&lt;service&gt;</td>
<td>Contains a &lt;port&gt; child element that describes the URL where the service is located. This is the location of the ultimate web service.</td>
</tr>
<tr>
<td>&lt;binding&gt;</td>
<td>Defines the message format and protocol details for each port. The binding element has two attributes: the name attribute and the type attribute. This element specifies how the client and the web service should send messages to one another.</td>
</tr>
</tbody>
</table>

Note: For more information about the WSDL standard, refer to the World Wide Web Consortium (W3C) Website at [w3.org/TR/wsdl](http://w3.org/TR/wsdl).

### 1.3.3 SOAP

- Is a simple XML-based protocol that allows applications to exchange information over HTTP.
- Is built on open standards supported by numerous development tools on various platforms.
- Is a request interface object in your application programming language.
- Provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.
- Enables the data to pass through layers of intermediaries and arrive at the ultimate receiver the way it was intended.

Note: You may not need to actually construct the SOAP messages yourself – many development tools available today construct SOAP behind the scenes.
1.3.3.1 SOAP Message

A SOAP message is an XML document that can be a request for a web service from a client or a “reply” from a web service to a client.

- Required <SOAP:Envelope>
- Optional <SOAP:Header>
- Required <SOAP:Body>

1.3.3.1.1 Example: Delete Tag Request (SOAP Message)

```xml
<SOAP-ENV:Envelope
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns="http://fedex.com/ws/ship/v13">
 <SOAP-ENV:Body>
  <DeleteTagRequest>
   <WebAuthenticationDetail>
    <UserCredential>
     <Key>
      User Key
     </Key>
     <Password>
      User Password
     </Password>
    </UserCredential>
    </WebAuthenticationDetail>
   <ClientDetail>
    <AccountNumber>xxxxxxxxx</AccountNumber>
    <MeterNumber>xxxxxx</MeterNumber>
   </ClientDetail>
   <Version>
    <ServiceId>ship</ServiceId>
    <Major>12</Major>
    <Intermediate>0</Intermediate>
    <Minor>0</Minor>
   </Version>
   <DispatchLocationId>MQYA</DispatchLocationId>
   <DispatchDate>2012-06-01</DispatchDate>
   <Payment>
    <PaymentType>shipper</PaymentType>
    <Payor>
     <AccountNumber>xxxxxxxxx</AccountNumber>
    </Payor>
   </Payment>
   <ConfirmationNumber>997037200019454</ConfirmationNumber>
  </DeleteTagRequest>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.3.4 Non-SOAP Web Services

FedEx offers a non-SOAP web services solution that you can use to send transactions without having to use tools that provide SOAP protocol support for web services. This may be convenient for developers using environments that do not provide support for SOAP. With this interface, XML documents are sent directly to the FedEx servers via the HTTP POST command. FedEx provides a set of specifications and examples to help with the development of this type of communications method.

To use the non-SOAP web service solution, you must have a working knowledge of HTTPS and Secure Socket Layering (SSL) encryption, the ability to provide a secure SSL connection to FedEx and the ability to code to an
operation interface using XML.

The interfaces used in the SOAP and non-SOAP web services are defined in WSDL files. The WSDL files contain schemas that define the layout of the operations. The same WSDL file is used for both the SOAP and non-SOAP web service users.

Non-SOAP users are concerned only with the schema definitions and not the other WSDL components that are SOAP-specific. The XML data that is sent via the non-SOAP interface looks almost identical to the data that is sent via the SOAP interface. The only difference is that the data sent via the non-SOAP interface does not contain the wrapping Envelope and Body tags that are specific to SOAP. The following is an example of a TrackRequest using the non-SOAP interface.

1.3.4.1 Example Track Request

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <v9:TrackRequest>
      <v9:WebAuthenticationDetail>
        <v9:UserCredential>
          <v9:Key>xxxxxx</v9:Key>
          <v9:Password/>
        </v9:UserCredential>
      </v9:WebAuthenticationDetail>
      <v9:ClientDetail>
        <v9:AccountNumber>XXXX</v9:AccountNumber>
        <v9:MeterNumber>XXXX</v9:MeterNumber>
        <v9:Localization>
          <v9:LanguageCode>EN</v9:LanguageCode>
        </v9:Localization>
      </v9:ClientDetail>
      <v9:TransactionDetail>
        <v9:CustomerTransactionId>Track By Number_v9</v9:CustomerTransactionId>
        <v9:Localization>
          <v9:LanguageCode>EN</v9:LanguageCode>
        </v9:Localization>
      </v9:TransactionDetail>
      <v9:Version>
        <v9:ServiceId>trck</v9:ServiceId>
        <v9:Major>9</v9:Major>
        <v9:Intermediate>1</v9:Intermediate>
        <v9:Minor>0</v9:Minor>
      </v9:Version>
      <v9:SelectionDetails>
        <v9:CarrierCode>FDXE</v9:CarrierCode>
        <v9:PackageIdentifier>
          <v9:Type>TRACKING_NUMBER_OR_DOORTAG</v9:Type>
          <v9:Value>XXXX</v9:Value>
        </v9:PackageIdentifier>
        <v9:ShipmenAccountNumber/>
      </v9:SelectionDetails>
      <v9:Destination>
        <v9:StreetLines>Address_Line</v9:StreetLines>
        <v9:City>City</v9:City>
        <v9:StateOrProvinceCode>XX</v9:StateOrProvinceCode>
        <v9:PostalCode>XXXX</v9:PostalCode>
      </v9:Destination>
      <v9:SelectionDetails>
      </v9:SelectionDetails>
    </v9:TrackRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
1.3.4.2 Error Handling

Error handling for non-SOAP operations is different from error handling for SOAP operations. The SOAP specification provides an error handling mechanism that is not present for non-SOAP operations. For a SOAP operation, a fault is returned as a SOAP exception. For a non-SOAP request, the contents of the SOAP fault are returned as an XML document. These SOAP fault documents are returned in situations such as schema validation failures or when operation types are unrecognized. In the following example, a SOAP fault document is returned from a schema validation failure in which the AccountNumber element was incorrectly sent as the AccountNumberx element:

```xml
<soapenv:Fault xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <faultcode>soapenv:Server</faultcode>
  <faultstring>5: Schema validation failed for request.</faultstring>
  <detail>
    <con:fault xmlns:con="http://www.bea.com/wli/sb/context">
      <con:errorCode>5</con:errorCode>
      <con:reason>Schema validation failed for request.</con:reason>
      <con:details>
        <con1:ValidationFailureDetail
          xmlns:con1="http://www.bea.com/wli/sb/stages/transform/config">
          <con1:message>Expected element 'AccountNumber@http://fedex.com/ws/ship/v8' instead of 'AccountNumberx@http://fedex.com/ws/ship/v8' here in element ClientDetail@http://fedex.com/ws/ship/v8</con1:message>
          <con1:xmlLocation>
            <ship:AccountNumberx
          </con1:xmlLocation>
          <con1:message>Expected element 'AccountNumber@http://fedex.com/ws/ship/v1' before the end of the content in element ClientDetail@http://fedex.com/ws/ship/v8</con1:message>
          <con1:xmlLocation>
            <ship:ClientDetail
              xmlns:ship="http://fedex.com/ws/ship/v8">
              <ship:AccountNumberx>000000000000000000</ship:AccountNumberx>
              <ship:MeterNumber>0000000</ship:MeterNumber>
            </ship:ClientDetail>
          </con1:xmlLocation>
        </con1:ValidationFailureDetail>
        <con:location>
          <con:node>Validate</con:node>
          <con:pipeline>Validate_request</con:pipeline>
          <con:stage>ValidateRequest</con:stage>
          <con:path>request-pipeline</con:path>
        </con:location>
      </con:details>
    </con:fault>
  </detail>
</soapenv:Fault>
```

Each reply should be checked for the Fault element to indicate failure in processing the message.

*Note: Normal error processing still applies; this is an additional error check for incorrect syntax in XML documents.*

Keep in mind that if you use either the SOAP or non-SOAP version of FedEx Web Services, labels are returned as Base64 encoded. To print shipping labels, you must decode labels before sending them to your printer.

1.3.4.3 Non-SOAP HTTP POST Example

The following HTTPS POST example is a valid working example, but is not guaranteed to work for all programming languages, applications, and host systems:

```http
POST /xml HTTP/1.0
Referrer: YourCompanyNameGoesHere
Host: ws.fedex.com
Port: 443
Accept: image/gif, image/jpeg, image/pjpeg, text/plain, text/html, */*
Content-Type: text/xml
```
Your FedEx Transaction

Each line is followed by one new line character except Content-length  and the FedEx transaction. Two new line characters follow the Content-length line. The FedEx transaction has no extra characters. The Content-length line should have the length of the FedEx transaction in place of the %d variable.

Note: Port 443 must be opened for bi-directional communication on your firewall.

After formatting your non-SOAP transaction and placing it in a HTTP POST request, you will need to open an SSL connection to the FedEx test server and send the request through FedEx by using your SSL connection. Next, parse the HTTPS response to determine if there were any errors. Examine the HTTP header to determine if any HTTP or Web Server errors were encountered. If you received a 200 status code, parse the reply to determine if there were any processing problems.

1.3.5 Visual Basic Project Error

You may receive an error indicating that an element is not set, even after setting it in the code. When you set a Boolean type element to true, you may also need to set the specified element to true.

1.4 Implementing FedEx Web Services

Before you begin implementing FedEx Web Services, note the following guidelines:

- FedEx Web Services are designed for use by skilled developers who are familiar with the communication standards SOAP and Web Services Description Language (WSDL).
- Unlike traditional client/server models, such as a web server or web page system, web services do not provide the user with a graphical user interface (GUI). Instead, web services share business logic, data, and processes through a programmatic interface across a network.
- To perform a particular FedEx task such as tracking a package, you need to use a class, module, or function that creates your request, sends it to the FedEx platform, and handles the response.
- FedEx Web Services are designed to support any operating system and coding language. Downloadable sample code is available in Java, C#, VB, .Net and PHP languages from the FedEx Developer Resource Center Technical Resources.
- Transactions submitted to FedEx using FedEx Web Services are required to have a minimum of 128-bit encryption to complete the request.

1.5 Understanding the XML Schema

The XML schema defines the messages that you can use to access the FedEx services. You create a request that contains business data and other instructions and you send it to FedEx. FedEx replies with a response that contains the data resulting from the instructions you sent in.

Note: The schema diagrams are conveniently linked to help you find information and child values.

The XML schema provides a means for defining the structure, content, and semantics of XML documents. An XML schema defines:

- Elements and attributes that can appear in a document
- Elements that are child elements
- Order and number of child elements
- Whether an element is empty or can include text
- Data types, default values, and fixed values for elements and attributes

Some important facts about the XML schema:

- Elements that contain sub-elements or carry attributes have complex types.
- Elements that contain numbers (and strings, and dates, etc.), but do not contain any sub-elements, have simple types. Some elements have attributes. Attributes always have simple types.
- Complex types in the instance document, and some of the simple types, are defined in the schema
1.5.1 Guide to the XML Schema

The XML schema for each WSDL provides details about the structure, content, and semantics of the request XML document sent to a FedEx Web Service and the XML document returned by that FedEx Web Service.

The top of each service schema includes:

- Schema location and schema file name that ends in an ".xsd" suffix.
- Alphabetical listing of complex types for the documented service.
- Alphabetical listing of schema simple types for the documented service.
- Input or request data type for the documented service.
- Output or reply data type for the documented service.

The remainder of the service schema contains tables of information about each element, complex type, and simple type.

Each table consists of some or all of the following sections: diagram, namespace, children, type, properties, used by, facets, and source.

1.6 Implementation Process

Planning your integration and organizing your application data to address your shipping needs can sometimes take more time than the actual implementation of the integration. FedEx Web Services conform to industry standards and are compatible with a comprehensive array of developers’ tools. This ensures the fastest time-to-market with maximum flexibility to integrate FedEx transactions and information into your applications. FedEx WSDLs are fully interoperable with any product or developer’s tool that also conforms to the WS-I Basic Profile. For details, see ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.

To obtain FedEx Web Services and begin integrating with an application, you need to access documentation, sample code, and sample service requests and replies with the WSDLs from the FedEx Developer Resource Center Technical Resources. Also, obtain a test meter number to engage in real-time online testing in the FedEx hosted test environment.

Note: Not all services are available outside the U.S.

1.6.1 Testing

FedEx supplies a complete online operating environment with which to test your applications against live FedEx servers. To execute test interactions, you must first include a test account number, test meter number, authentication key, and password in your code. These credentials are provided to registered developers at the FedEx Developer Resource Center at www.fedex.com/developer.

Production credentials can be obtained prior to the certification process. Advanced services are not enabled, but standard services are enabled. Refer to Preproduction Assistance for more information on support from FedEx.

1.6.1.1 Preproduction Assistance

Preproduction assistance is available via the FedEx Web Integrated Solutions Consultation (WISC) team. If you are in the preproduction stages of implementing a FedEx web integrated solution and would like to speak with a FedEx integration consultant who can assist you in understanding FedEx Web Services, contact your FedEx sales executive or technical support at 1.877.339.2774 Monday thru Friday, 7 a.m. to 9 p.m. and Saturday 9 a.m. to 3 p.m. (CST). Both your FedEx sales executive and technical support can request a WISC team member to contact you within 3 business days.
Corporate developers may find that solutions to their needs have already been implemented by a software vendor that is FedEx® Compatible. If improved time-to-market, cost containment, or specialized knowledge is needed, corporate development planners may want to review the available third-party solutions. To see a list of the solutions provided by the FedEx® Compatible providers, go to the Available FedEx® Compatible Solutions page at http://www.fedex.com/us/compatible/.

1.6.2 Certification

Certification is the process of ensuring that your implementation meets a number of requirements for safe, secure, and effective operation of your solution in the FedEx production environment. Certification requirements differ based on whether you are a corporate or commercial developer, and whether you are implementing using the advanced or standard services. Certification is not required for rating and tracking in production.

1.6.3 Go To Production

Once an application has passed certification, the developer must replace the test credentials with the production credentials issued by FedEx. The application connection is then directed to the production servers, and the application is live.

1.6.3.1 Requirements for Corporate and Non-Commercial Developers

There are some differences in how support is provided and in the approvals required to go into production that depend on whether you are creating an application for use by your own company or if you are planning to resell your solution to others.

1.6.3.2 Requirements and Resources for Corporate Developers

Corporate developers are typically part of a dedicated development team at a single company. This category also includes third-party developers (consultants) hired by the company to work on its behalf. In all cases, the integration will be used by the company itself and will not be resold or distributed outside of its own footprint. In this situation, FedEx can support the customer directly.

Table 2. Requirements and Resources for Corporate Developers

<table>
<thead>
<tr>
<th>Requirements and Resources for Corporate Developers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Must be accepted into the FedEx® Compatible</td>
<td>No</td>
</tr>
<tr>
<td>Self-certification of implementations using standard services</td>
<td>Yes</td>
</tr>
<tr>
<td>Self-certification of implementations using advanced services</td>
<td>No</td>
</tr>
<tr>
<td>Certification assistance</td>
<td>Yes (WISC team)</td>
</tr>
<tr>
<td>FedEx supports the customer directly</td>
<td>Yes</td>
</tr>
</tbody>
</table>

1.6.3.2.1 Requirements for Consultants

Consultants developing on behalf of a corporate customer must ensure that their client provides their account information and a signed End User License Agreement (EULA) to FedEx to obtain a production test meter.

1.6.3.2.2 Requirements and Resources for Commercial Developers

Commercial developers create solutions with the intent of distributing and/or reselling them to their customers. Because they are deployed in a variety of situations, commercial integrations generally require a higher order of
“fit and finish.” Commercial developers are responsible for supporting their products for their customers. FedEx has a dedicated team of professionals to help developers commercialize their products and to coordinate the three-way interplay between the developer, the end customer, and FedEx.

If you are a commercial developer interested in becoming a FedEx Compatible provider, go to http://www.fedex.com/us/compatible/ for more information about the FedEx Compatible Program.

### 1.6.3.3 URL Errors

If a VB.NET or C# project still sends transactions to the test server after changing the URL in the WSDLs to print to production, perform the following:

- Make sure permissions are already activated in the production environment.
- Copy the WSDL files to a different folder.
- Follow the directions on changing the new WSDL files to point to production, as described in the FedEx Developer Resource Center in the “Move to Production” topic.
- Remove existing web services references from your project that point to old WSDLs containing the URLs to the test environment.
- Create new web references that point to the modified WSDLs. Use the same names as the old references.
- Compile and test the project. Your new production credentials should work for standard web services, such as rating or tracking without extra permissions. Advanced web services require permissions to be active before they will work. Old test key values will now return an error message. Test keys will no longer work with production server addresses.
8. Address Validation Service

Use the Address Validation Service (AVS) to validate or complete recipient addresses. This service validates recipient addresses before you ship packages, provides descriptive error details and corrected options for invalid addresses, and/or determines whether an address is business or residential to increase the accuracy of courtesy rate quotes.

Note: Do not use this service to determine the deliver-ability of an address. FedEx does offer delivery service to every valid address. However, FedEx does not deliver to PO Boxes (except via SmartPost).

8.1 Address Validation Request

The AddressValidation WSDL AddressValidationRequest allows you to validate recipient address information before you ship a package. Correct addresses on the shipping label will help eliminate delivery delays and additional service fees.

Note:
- The Address Validation Service is an advanced service and must be enabled by FedEx Customer Support for production use. Contact your FedEx account executive for more information.
- Address resolution rates vary by country.

Table 54. Countries where Address Validation works

<table>
<thead>
<tr>
<th>Countries where Address Validation works</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Virgin Islands</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>Brazil</td>
<td>Switzerland</td>
</tr>
<tr>
<td>Mexico</td>
<td>Austria</td>
</tr>
<tr>
<td>Bahamas</td>
<td>Sweden</td>
</tr>
<tr>
<td>Cayman Islands</td>
<td>Estonia</td>
</tr>
<tr>
<td>Argentina</td>
<td>Finland</td>
</tr>
<tr>
<td>Aruba</td>
<td>Greece</td>
</tr>
<tr>
<td>Barbados</td>
<td>Norway</td>
</tr>
<tr>
<td>Bermuda</td>
<td>Portugal</td>
</tr>
<tr>
<td>Chile</td>
<td>South Africa</td>
</tr>
<tr>
<td>Costa Rica</td>
<td>Panama</td>
</tr>
<tr>
<td>Dominican Republic</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>Guatemala</td>
<td>Uruguay</td>
</tr>
<tr>
<td>Jamaica</td>
<td>Venezuela</td>
</tr>
<tr>
<td>Netherlands</td>
<td>Columbia</td>
</tr>
<tr>
<td>Antilles</td>
<td>France</td>
</tr>
<tr>
<td>Germany</td>
<td>Peru</td>
</tr>
<tr>
<td>Spain</td>
<td>Singapore</td>
</tr>
</tbody>
</table>
Use the Address Validation request to perform the following:

- Complete incomplete recipient addresses. For example, the AVS helps in completing incomplete recipient address information like zip code, but not missing apartment number. In some cases, AVS may be able to add missing information, depending on the verification of the provided information against reference data. AVS cannot add missing secondary information (i.e. apartment or suite) at this time.
- Correct invalid recipient addresses.
- Determine whether an address is business or residential to increase the accuracy of courtesy rate quotes. Applies to U.S. and Canada addresses only.
- Confirm the validity and completeness of addresses in many countries in these regions - U.S., Canada, Latin America, Europe and Middle East and Asia Pacific. You are now able to validate domestic and international address information in the following countries before shipping a package, eliminating unnecessary delivery delays and additional service fees.

Note: More countries will be supported throughout the year.

Legal Disclaimer:
The data provided herein is FedEx proprietary and confidential information, provided as a courtesy at your request. No part of this data may be distributed or disclosed in any form to any third party without the written permission of FedEx. It reflects the current FedEx address-level business/residential classification in the FedEx delivery address database, and is subject to change. In furnishing this information, FedEx does not guarantee its present or future accuracy, and does not guarantee that packages shipped to these addresses will be invoiced according to the business/residential classification provided herein. Providing this information shall not be deemed to alter the terms of the relationship between the parties. See the FedEx Service Guide and any applicable account pricing agreement for terms and conditions governing FedEx shipping and charges.

8.2 Address Validation Service Details

The following service details apply to Address Validation:

- Provides street level matches.
- Receives monthly updates to its address database.
- Checks addresses in several countries in these regions - U.S., Canada, Latin America, Europe and Middle East and Asia Pacific.
- Can distinguish between business and residential addresses (for U.S. and Canada only).
- Does not match addresses based upon individual/personal names.
- FedEx does not normally deliver to P.O. box addresses for U.S. or U.S. inbound shipments. However, FedEx may deliver to post office boxes in some rural locations if the P.O. box is associated with an address. You may also use P.O. box addresses for certain international locations, including shipments to Puerto Rico, but you must include a valid phone, fax or telex number on the label.
- Up to 100 addresses can be checked in one Web Service request.

For more detailed information about the services offered by FedEx, see the electronic FedEx Service Guide.
8.3 How FedEx Address Validation Works

- Checks if street, city, state or province, or postal code is entered. In some cases, AVS may be able to add missing information, depending on the verification of the provided information against reference data. AVS cannot add missing secondary information (i.e. apartment or suite) at this time.
- Checks if the street number is within a valid range for the street entered.
- Informs you when an address cannot be resolved (address verified against reference data) based on the street number, street name, city, state or province, or postal code entered.

8.4 Tips on Getting Accurate Address Matches

Use correct spacing: Make sure spaces are placed correctly and avoid unnecessary spaces.

Use correct spelling: Eliminate spelling and typographic errors. Make sure you have the correct usage of the number zero (0) and letter O.

Avoid special characters: Refrain from using special characters not required for the address, such as periods after abbreviations (Ave vs. Ave.)

Provide additional address and street information: Providing additional address information can increase the accuracy of address results. For example:

- Building or house number such as 1, 1A, One
- Street name such as Main, George Washington, 42nd
- Street Suffix such as Road, Avenue, Rd, Ave

Enter city, state/province and postal code: Providing all address information will increase the accuracy of your results. The ZIP+4 portion of the postal code is not necessary to check an address.

Use correct abbreviations: The United States Postal Service and postal authorities in other countries define standard abbreviations for state/province, street suffix, and apartment/unit designations. A nonstandard abbreviation may cause poor search results. If you are unsure about an abbreviation, do not use it.

8.5 Address Results

Urbanization (Puerto Rico only): This descriptor, commonly used in urban areas of Puerto Rico, is an important part of the address format as it describes the location of a given street. In Puerto Rico, repeated street names and address number ranges can be found within the same postal code. These streets can have the same house number ranges. In these cases, the urbanization name is needed to correctly identify the location of a particular address.

For example:
Sr Pedro Rivera Urb Hermosillo 123 Calle 1 Bayamon, PR 00961-1212

8.6 Address Checking Process

Address Validation Service (AVS) is the new central database for address-related information across FedEx. Address Validation returns the "best" address in the response—one that has been validated against reference data.

Address Validation Service rules:

- If too many changes are required, AVS will not be able to find a match to reference data. In this situation, AVS will return a formatted form of the raw address input.
- If just a zip code is provided, Address Validation Service returns 'unknown' for the business/residential classification.
- Four address classification values may be returned--Business, Residential, Mixed or Unknown.
8.7 Address Classification

Address Validation uses reference data to determine the classification of a given address. The classification is calculated as part of the address validation process. The classification for a functional address is calculated independently of the address validation process and is based on feedback by operational personnel, with commercial data sources used for confirmation only.

Address Validation has only four possible classifications for addresses: unknown, business, residential and mixed. All addresses begin with an “unknown” classification and stay that way until Address Validation business rules determine that their classifications should change. A location only gets a “mixed” classification if it is a multi-tenant based address and contains both business and residential units.

8.7.1 Residential Address Classification

Residential address relates to a home or private residence, including locations where a business is operated from the home.

8.7.2 Address Matching Classification

Table 55. Address Matching Classification

<table>
<thead>
<tr>
<th>Address Matching Classification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolved</td>
<td>The input address was matched with an acceptable level of confidence to a record in a reference data set. Note that the level to which an address has been resolved is described separately by other flags and indicators. See notes.</td>
</tr>
<tr>
<td>Not Resolved</td>
<td>The input address was not matched to a reference data set, but it was parsed and normalized (standard abbreviations applied).</td>
</tr>
<tr>
<td>Country Not Supported</td>
<td>Address Validation Service does not currently include reference data to support the country of the input address. The raw address is stored and a Address Validation Service ID is assigned, but no additional processing is applied.</td>
</tr>
<tr>
<td>Country Unknown</td>
<td>The country of the address could not be determined. The raw address is stored and a Address Validation Service ID is assigned, but no additional processing is applied.</td>
</tr>
<tr>
<td>Not Processed</td>
<td>The address could not be processed because of internal errors. The raw address is stored and a Address Validation Service ID is assigned, but no additional processing is applied.</td>
</tr>
<tr>
<td>Blank</td>
<td>The input address in the request contained no data (blank). No data will be stored in Address Validation Service and no Address Validation Service ID will be assigned.</td>
</tr>
</tbody>
</table>

8.7.3 Address Formats

- **RAW**: Address information provided by the user. The Raw address is returned when the Address Validation Service does not support the country for address validation.
- **NORMALIZED**: A formatted version of the address where elements are parsed and standard abbreviations are applied. The Normalized address is returned when the Address Validation Service supports a country for address validation, but cannot match the address against reference data. Reference data include postal data (and map data, for the US only).
- **STANDARDIZED**: A formatted and validated version of the address. The standardized address is returned when the Address Validation Service can match the address against reference data. Note that the Address Validation Service may make slight changes to the address in order to find a match.
8.8 Address Validation Coding Details

The following information is the minimum required to check an address:

- Address with at least 1 address line
- City
- State/Province (if applicable)
- Postal (if country is postal aware)
- Country

8.8.1 AddressValidationRequest Elements

Table 56. AddressValidationRequest Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressValidationRequest</td>
<td>Validation: Given a raw address the AddressValidation system responds with all of the information it can determine about that address including the Business Residential classification that was in effect at the time, including information on how the classification was calculated.</td>
</tr>
<tr>
<td>AddressValidationRequest/InEffectAsOfTimestamp</td>
<td>Optional DateTime field used to request the data as of this point in time. This defaults to current date time (of the AddressValidation System). This is useful because the AddressValidation database is dynamic and stores historical data. Characteristics such as Business/Residential indicator may change over time. Eg. 2013-01-11 T 07:52:56</td>
</tr>
<tr>
<td>AddressToValidate</td>
<td></td>
</tr>
<tr>
<td>AddressToValidate/ClientReferenceId</td>
<td>A reference id provided by the client.</td>
</tr>
<tr>
<td>AddressToValidate/Contact</td>
<td>The descriptive data for a point-of-contact person</td>
</tr>
<tr>
<td>Contact/ContactId</td>
<td>Client provided identifier corresponding to this contact information.</td>
</tr>
<tr>
<td>Contact/PersonName</td>
<td>Optional. Identifies the contact person's name.</td>
</tr>
<tr>
<td>Contact/Title</td>
<td>Identifies the contact person's title.</td>
</tr>
<tr>
<td>Contact/CompanyName</td>
<td>Optional. Identifies the company this contact is associated with.</td>
</tr>
<tr>
<td>Contact/PhoneNumber</td>
<td>Identifies the phone number associated with this contact.</td>
</tr>
<tr>
<td>Contact/PhoneExtension</td>
<td>Identifies the phone extension associated with this contact.</td>
</tr>
<tr>
<td>Contact/TollFreePhoneNumber</td>
<td>Identifies a toll free number, if any, associated with this contact.</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Contact/PagerNumber</td>
<td>Identifies the pager number associated with this contact.</td>
</tr>
<tr>
<td>Contact/FaxNumber</td>
<td>Identifies the fax number associated with this contact.</td>
</tr>
<tr>
<td>Contact/EMailAddress</td>
<td>Identifies the email address associated with this contact.</td>
</tr>
<tr>
<td>AddressToValidate/Address</td>
<td>Descriptive data for a physical location. May be used as an actual physical address (place to which one could go), or as a container of “address parts” which should be handled as a unit (such as a city-state-ZIP combination within the US).</td>
</tr>
<tr>
<td>Address/StreetLines (0 to 4 repetitions)</td>
<td>Combination of number, street name, etc. At least one line is required for a valid physical address; empty lines should not be included.</td>
</tr>
<tr>
<td>Address/City</td>
<td>Required. Name of city, town, etc.</td>
</tr>
<tr>
<td>Address/StateOrProvinceCode</td>
<td>Required. Identifying abbreviation for US state, Canada province, etc. Format and presence of this field will vary, depending on country.</td>
</tr>
<tr>
<td>Address/PostalCode</td>
<td>Required. Identification of a region (usually small) for mail/package delivery. Format and presence of this field will vary, depending on country.</td>
</tr>
<tr>
<td>Address/UrbanizationCode</td>
<td>Optional. Relevant only to addresses in Puerto Rico.</td>
</tr>
<tr>
<td>Address/CountryCode</td>
<td>The two-letter code used to identify a country.</td>
</tr>
<tr>
<td>Address/CountryName</td>
<td>Required. The fully spelt out name of a country.</td>
</tr>
<tr>
<td>Address/Residential</td>
<td>Indicates whether this address residential (as opposed to commercial).</td>
</tr>
</tbody>
</table>

### 8.8.2 AddressValidationReply Elements

Any error conditions or address-checking issues are returned in the Address Validation reply. The following table describes Address Validation reply elements:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplyTimestamp</td>
<td>Date and time</td>
</tr>
<tr>
<td>AddressValidationResult</td>
<td></td>
</tr>
<tr>
<td>AddressValidationResult/ClientReferenceId</td>
<td>The client reference id for the validated address.</td>
</tr>
<tr>
<td>AddressValidationResult/State</td>
<td>Specifies the degree to which service was able to simplify the address provided, as per USPS standards and match it to an address already in the internal FedEx address repository.</td>
</tr>
<tr>
<td>AddressValidationResult/FedExAddressClassificationType</td>
<td>UNKNOWN, BUSINESS, RESIDENTIAL, MIXED</td>
</tr>
<tr>
<td><strong>AddressValidationResult/EffectiveContact</strong></td>
<td>The descriptive data for a point-of-contact person</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>EffectiveContact/ContactId</strong></td>
<td>Client provided identifier corresponding to this contact information.</td>
</tr>
<tr>
<td><strong>EffectiveContact/PersonName</strong></td>
<td>Identifies the contact person's name.</td>
</tr>
<tr>
<td><strong>EffectiveContact/Title</strong></td>
<td>Identifies the contact person's title.</td>
</tr>
<tr>
<td><strong>EffectiveContact/CompanyName</strong></td>
<td>Identifies the company this contact is associated with.</td>
</tr>
<tr>
<td><strong>EffectiveContact/PhoneNumber</strong></td>
<td>Identifies the phone number associated with this contact.</td>
</tr>
<tr>
<td><strong>EffectiveContact/PhoneExtension</strong></td>
<td>Identifies the phone extension associated with this contact.</td>
</tr>
<tr>
<td><strong>EffectiveContact/TollFreePhoneNumber</strong></td>
<td>Identifies a toll free number, if any, associated with this contact.</td>
</tr>
<tr>
<td><strong>EffectiveContact/PagerNumber</strong></td>
<td>Identifies the pager number associated with this contact.</td>
</tr>
<tr>
<td><strong>EffectiveContact/FaxNumber</strong></td>
<td>Identifies the fax number associated with this contact.</td>
</tr>
<tr>
<td><strong>EffectiveContact/EMailAddress</strong></td>
<td>Identifies the email address associated with this contact.</td>
</tr>
<tr>
<td><strong>AddressValidationResult/EffectiveAddress</strong></td>
<td>Descriptive data for a physical location. May be used as an actual physical address (place to which one could go), or as a container of &quot;address parts&quot; which should be handled as a unit (such as a city-state-ZIP combination within the US).</td>
</tr>
<tr>
<td><strong>EffectiveAddress/StreetLines</strong></td>
<td>Combination of number, street name, etc. At least one line is required for a valid physical address; empty lines should not be included.</td>
</tr>
<tr>
<td><strong>EffectiveAddress/City</strong></td>
<td>Name of city, town, etc.</td>
</tr>
<tr>
<td><strong>EffectiveAddress/StateOrProvinceCode</strong></td>
<td>Identifying abbreviation for US state, Canada province, etc. Format and presence of this field will vary, depending on country.</td>
</tr>
<tr>
<td><strong>EffectiveAddress/PostalCode</strong></td>
<td>Identification of a region (usually small) for mail/package delivery. Format and presence of this field will vary, depending on country.</td>
</tr>
<tr>
<td><strong>EffectiveAddress/UrbanizationCode</strong></td>
<td>Relevant only to addresses in Puerto Rico.</td>
</tr>
<tr>
<td><strong>EffectiveAddress/CountryCode</strong></td>
<td>The two-letter code used to identify a country.</td>
</tr>
<tr>
<td><strong>EffectiveAddress/CountryName</strong></td>
<td>The fully spelled out name of a country.</td>
</tr>
<tr>
<td><strong>EffectiveAddress/Residential</strong></td>
<td>Indicates whether this address residential (as opposed to commercial).</td>
</tr>
<tr>
<td><strong>AddressValidationResult/ParsedAddressPartsDetail</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ParsedAddressPartsDetail/ParsedPostalCode</strong></td>
<td>The postal code specified in a form that is</td>
</tr>
<tr>
<td>ParsedAddressPartsDetail/ParsedStreetLineDetail</td>
<td>supported by USPS as base, secondary and tertiary.</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/houseNumber</td>
<td>House Number resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/preStreetType</td>
<td>Pre Street Type resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/leadingDirectional</td>
<td>Leading Directional resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/streetName</td>
<td>Street Name resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/streetName2</td>
<td>Street Name 2 resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/streetSuffix</td>
<td>Street Suffix resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/trailingDirectional</td>
<td>Trailing Directional resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/unitLabel</td>
<td>Unit Label resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/unitNumber</td>
<td>Unit Number resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/subUnitLabel</td>
<td>SubUnit Label resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/subUnitNumber</td>
<td>SubUnit Number resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/deliveryStation</td>
<td>DeliveryStation resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>deliveryStationNumber</td>
<td>DeliveryStation Number resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/RuralRoute</td>
<td>Rural Route (RR) / Highway Contract (HC)</td>
</tr>
<tr>
<td>ParsedStreetLineDetail/PoBox</td>
<td>PO Box resulting from standardization/normalization</td>
</tr>
<tr>
<td>parsedStreetLineDetail/Building</td>
<td>process, when relevant</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>parsedStreetLineDetail/Organization</td>
<td></td>
</tr>
<tr>
<td>parsedAddressParts/BuildingComplete</td>
<td>BuildingComplete resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>parsedStreetLineDetail/subBuildingComplete</td>
<td>SubBuildingComplete (label and number) resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>parsedStreetLineDetail/subBuildingComplete2</td>
<td>SubBuilding Complete2 (label and number) resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>parsedAddressParts/OrganizationComplete</td>
<td>Organization Complete (number and name) resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>parsedStreetLineDetail/organizationComplete2</td>
<td>Organization Complete 2 (number and name) resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>parsedStreetLineDetail/rrhc</td>
<td>RRHC (rural route highway contract) resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>parsedStreetLineDetail/privateMailbox</td>
<td>Private Mailbox resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>city</td>
<td>City (Locality1) resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>Locality2</td>
<td>Locality2 resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>Locality3</td>
<td>Locality3 resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>stateProvince</td>
<td>State (Province) resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>stateProvince2</td>
<td>Province2 resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>parsedPostalCodeDetail/Base</td>
<td>US Postal Code Base</td>
</tr>
<tr>
<td>parsedPostalCodeDetail/AddOn</td>
<td>US Postal Code AddOn</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>ParsedPostalCodeDetail/DeliveryPoint</code></td>
<td>US Postal Code Delivery Point: value from 00 to 99</td>
</tr>
<tr>
<td><code>postalCode/postalCodeBase</code></td>
<td>Postal Code Base resulting from standardization/normalization process, when relevant (5-digit US, complete postal code for International)</td>
</tr>
<tr>
<td><code>postalCode/postalCodeAddOn</code></td>
<td>Postal Code Add On resulting from standardization/normalization process, when relevant (US Only, only +4)</td>
</tr>
<tr>
<td><code>postalCode/postalCodeBarCode</code></td>
<td>Postal Code Bar Code resulting from standardization/normalization process, when relevant (US Only, only +2)</td>
</tr>
<tr>
<td><code>postalText</code></td>
<td>Formatted Postal Code resulting from standardization/normalization process, when relevant. For US this is the complete, formatted postal code (including hyphens) Internationally this is the complete, formatted postal code (following country-specific rules). For international addresses the postText value is the same as the PostalCodeBase value.</td>
</tr>
<tr>
<td><code>country</code></td>
<td>Country resulting from standardization/normalization process. When no country is provided in the Raw, SHARE may insert a country (Country Resolver Logic) or leave blank.</td>
</tr>
</tbody>
</table>
| `parentId` | ID of Standardized Address stanza that is related to this normalized address (if applicable)  
  *Note: This field is currently being populated via a crawler (started Oct 2014)* |
<p>| <code>effDate</code> | Date/time when standardization/Normalized Address stanza was created and/or updated via maintenance. |
| <code>lastRefDate</code> | Date/time when standardization/Normalized Address was last referenced. Note: This is for internal use only as is not updated real-time (lag of 30-90 days can be expected). |
| <code>AddressValidationResult/AddressAttribute</code> | Specifies additional information about the address processed by the system as a key-value pair. |
| <code>AddressAttribute/Name</code> | Specifies the key for the address attribute. |
| <code>AddressAttribute/Value</code> | The value for the key for address attribute |
| <code>shareId</code> | Unique key for address. Points to best version of address. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>operationalAddressId</td>
<td>Operational address ID for best address. Includes address ID + NULL</td>
</tr>
<tr>
<td>effectiveDateTime</td>
<td>Date/time when SHARE ID/Operational Address stanza was created.</td>
</tr>
<tr>
<td></td>
<td><em>Note: If the address stanza is deleted (i.e. during a purge) and then reprocessed, the effective date will reflect the date/time that the reprocessing occurred.</em></td>
</tr>
<tr>
<td>addressType</td>
<td>Operational address type (STANDARDIZED, NORMALIZED, or RAW)</td>
</tr>
<tr>
<td>addressState</td>
<td>Operational address state (RESOLVED, NOT_RESOLVED, NOT_PROCESSED)</td>
</tr>
<tr>
<td>addressId</td>
<td>Address ID for standardized/normalized/operational address</td>
</tr>
<tr>
<td>line (up to four lines)</td>
<td>Address lines resulting from standardization/normalization process, when relevant</td>
</tr>
<tr>
<td>AddressType</td>
<td>Indicates type of address stanza (STANDARDIZED/NORMALIZED)</td>
</tr>
<tr>
<td>Inserted</td>
<td>Indicates if Standardized/Normalized Address Stanza was inserted into SHARE database</td>
</tr>
<tr>
<td>BuildingValidated</td>
<td>Indicates if the Building was validated against reference data</td>
</tr>
<tr>
<td>Confidence</td>
<td>Calculated value indicating the amount of change that must be introduced to be able to standardize the address. For US, the calculated value comes from a third-party vendor (currently, but could change). Internationally, SHARE calculates this value.</td>
</tr>
<tr>
<td>DataVintage</td>
<td>Data vintage used to evaluate the normalized address during the attempt to standardize</td>
</tr>
<tr>
<td>DPV</td>
<td>DPV=Delivery Point Valid. Indicator translated from values provided by the USPS that identify the validity of a postal delivery address. Provided for US addresses only that can be standardized against Postal Data. Not provided for US Geo Validated addresses</td>
</tr>
<tr>
<td>InvalidSuiteNumber</td>
<td>TRUE: Suite information was provided and was either incorrect, or was provided for an address that was not recognized as requiring secondary information</td>
</tr>
<tr>
<td></td>
<td>FALSE: Suite information was not provided and was not needed, or provided suite information was valid</td>
</tr>
<tr>
<td>Matched</td>
<td>TRUE = Matches to a record in the SHARE DB</td>
</tr>
<tr>
<td></td>
<td>FALSE = Does not match</td>
</tr>
<tr>
<td>MissingOrAmbiguousDirectional</td>
<td>Flag only returned when address is not</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MultipleMatches</td>
<td>For US Addresses: TRUE: More than one potential match to reference data is available, usually due to a simple difference, such as a leading directional. The address remains not resolved because there is not a systematic way to determine which candidate is appropriate. FALSE: No matches (not resolved), or a single match to reference data exists (resolved). For International Addresses: TRUE: More than one potential match to reference data is available (not specific to directional. It could be that another required address element is missing) FALSE: No matches (not resolved), or a single match to reference data exists (resolved).</td>
</tr>
<tr>
<td>OrganizationValidated</td>
<td>Indicates if the Organization was validated against reference data.</td>
</tr>
<tr>
<td>PostalValidated</td>
<td>Indicates if the PostalCode was validated against reference data. For US addresses, this is only returned when address cannot be standardized. Always returned for int'l addresses.</td>
</tr>
<tr>
<td>Resolved</td>
<td>Indicates if address can be standardized (resolved)</td>
</tr>
<tr>
<td>RRConversion</td>
<td>Indicates if a Rural Route conversion was applied to the address during standardization. This flag applies to Canadian and International addresses only. There is a similar flag (standardized.status.name = RRConversion) associated with the standardized address that applies to US addresses.</td>
</tr>
<tr>
<td>StreetValidated</td>
<td>Returned for Canada and Generic Resolver</td>
</tr>
<tr>
<td>StreetRangeValidated</td>
<td>TRUE: House number and street were validated against a range of house numbers for that street provided in the reference data. FALSE: House number and street were not validated.</td>
</tr>
<tr>
<td>SuiteRequiredButMissing</td>
<td>TRUE: indicates that an input address was resolved to a building base address and that a suite or unit number is required to achieve a more exact match, but this secondary address information is missing from the input address. FALSE: Indicates that a suite was either not needed and not provided, or was provided and</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>SuiteNotValidated</td>
<td>TRUE indicates: 1) input address contains suite information 2) reference data is available and has confirmed that this address is a building base 3) reference data is not available to validate the suite information FALSE indicates: 1) Either: a) Suite information was not provided as input b) Suite information was provided and reference data is available to validate the suite information</td>
</tr>
<tr>
<td>ValidlyFormed</td>
<td>Current Definition: Future Definition: TRUE indicates that sufficient information (the minimal elements needed to determine that the address meets the criteria of at least one address precision) has been provided to attempt to validate the address FALSE indicates that insufficient information has been provided to attempt to validate the address</td>
</tr>
<tr>
<td>Zip4Match</td>
<td>TRUE indicates that the input address was resolved to a standardized address based upon at least a ZIP+4 match. FALSE indicates that the address was not resolved to a standardized address based upon ZIP+4 match. (US Only)</td>
</tr>
<tr>
<td>Zip11Match</td>
<td>TRUE indicates that the input address was resolved to a standardized address based upon a match at the postal barcode level (i.e. Zip-11 match). This is the highest level of postal code validation. All addresses resolved with the ZIP-11 Match flag set will also have the ZIP-4 Match flag set. FALSE indicates that the input address was not resolved to a standardized address based upon Zip 11 match. (US Only)</td>
</tr>
<tr>
<td>AddressPrecision</td>
<td>Indicates the depth/precision of the address. MULTI_TENTANT_UNIT indicates that the address has valid secondary information. MULTI_TENTANT_BASE indicates that the address is a valid multi tenant location but secondary information either was not provided or could not be validated. STREET_ADDRESS indicates that the location is not a valid multi tenant location.</td>
</tr>
<tr>
<td>DataVintage</td>
<td>Month and year of the reference data that was used to identify the address</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EncompassingZIP</td>
<td>TRUE indicates that the current address’ zip code encompasses other zip codes. FALSE indicates that the current address’ zip code does not encompass other zip codes. (US only)</td>
</tr>
<tr>
<td>SplitZIP</td>
<td>TRUE when the address comes under a new ZIP code that did not previously exist. FALSE when the address does not come under a new ZIP code that did not previously exist.</td>
</tr>
<tr>
<td>UniqueZIP</td>
<td>TRUE indicates that the postal code of the address is unique for a specific postal customer address. The ZIP may apply to a single unit or floor within a building, to an entire building, or to an institution or corporate campus. FALSE indicates that the postal code of the address is not unique for a specific postal customer address.</td>
</tr>
<tr>
<td>InterpolatedStreetAddress</td>
<td>TRUE indicates that the house number of the address is valid within a known range of street numbers, but that the existence of the specific street number could not be confirmed. This usually occurs when postal data can’t confirm the address and mapping data is used instead. The house number of the address is included within the matched range, but the reference data does not include the point level address data required to validate that the input street number actually exists within the matched range.</td>
</tr>
<tr>
<td>Intersection</td>
<td>TRUE indicates that the address is an intersection. FALSE indicates that the address is not an intersection.</td>
</tr>
<tr>
<td>MatchSource</td>
<td></td>
</tr>
<tr>
<td>MultiUnitBase</td>
<td>TRUE indicates that an input address was resolved to a standardized address for the base address of a multi-unit building. FALSE indicates that the address was not resolved to a standardized address for the base address of a multi-unit building.</td>
</tr>
<tr>
<td>POBox</td>
<td>TRUE indicates that the input address was recognized as a PO Box address. FALSE indicates that the input address was not recognized as a PO Box address.</td>
</tr>
<tr>
<td>POBoxOnlyZIP</td>
<td>TRUE indicates that USPS considers this ZIP as a PO Box-only postal code. This means that USPS does not deliver to individual street addresses in the postal code. Valid street addresses may exist in the postal code, but they cannot be validated by the USPS reference data. FALSE indicates that the USPS does not</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PostalDataSource</td>
<td>Lists the postal data source for international address validation.</td>
</tr>
<tr>
<td>ResolutionMethod</td>
<td>Indicates type of resolution method utilized to resolve the address.</td>
</tr>
<tr>
<td>RRConversion</td>
<td>TRUE indicates that the input address was recognized as a Rural Route or Highway Contract addresses and that it was matched to a standardized address through a conversion to a normal street address. FALSE indicates that the input address was not recognized as a Rural Route or Highway Contract address and was not converted to a street address. (US only.)</td>
</tr>
<tr>
<td>RuralRoute</td>
<td>TRUE indicates that the input address was recognized as a Rural Route or Highway Contract addresses. FALSE indicates that the input address was not recognized as a Rural Route or Highway Contract address.</td>
</tr>
<tr>
<td>StreetAddress</td>
<td>TRUE indicates that the house number and street name were validated against reference data. FALSE indicates that the house number and street name were not validated against reference data. (Non-US addresses only, where applicable)</td>
</tr>
<tr>
<td>StreetRang</td>
<td>TRUE indicates that the address includes a street number range instead of a single house number. The range is from the input address from which this address was resolved, and that the input range was validated as being included within a known street range segment for the matched street. FALSE indicates that the address does not include a street number range. (Non-US addresses only, where applicable)</td>
</tr>
<tr>
<td>StreetPointNotValidated</td>
<td>TRUE indicates that the house number for the street address was not validated against reference data FALSE indicates that the house number for the street address was either not validated, not provided, or not relevant for the address (Non-US addresses only, where applicable)</td>
</tr>
<tr>
<td>StreetPointNotApplicable</td>
<td>TRUE indicates that house number at the street level is not applicable for this address FALSE indicates that the house number at the street level is applicable for this address (Non-US addresses only, where applicable)</td>
</tr>
<tr>
<td>StreetNameAddress</td>
<td>TRUE indicates that the street name was</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>StreetBuildingAddress</td>
<td>TRUE indicates that the building and street information were validated against reference data, but not house number. FALSE indicates that the building and street information were not validated against reference data. (Non-US addresses only, where applicable)</td>
</tr>
<tr>
<td>StreetOrganizationAddress</td>
<td>TRUE indicates that organization and street information were validated against reference data. FALSE indicates that organization and street information were not validated against reference data. (Non-US addresses only, where applicable)</td>
</tr>
<tr>
<td>ValidMultiUnit</td>
<td>TRUE indicates that the address includes a validated suite or unit number. FALSE indicates that the address does not include a validated suite or unit number.</td>
</tr>
</tbody>
</table>

### 8.8.3 Samples

Sample code, sample transactions and the specific individual WSDL/XML guide(s) can be downloaded following these steps:

1. Go to the FedEx Developer Resource Center (DRC) at [www.fedex.com/developer](http://www.fedex.com/developer). Log in with your FedEx.com username and password. If you do not have one, then click on the "Register Now" link. Fill out the form for your DRC username and password.

2. From the left navigation area of the page, click on the link 'FedEx Web Services'.

3. Click on the link 'Move to documentation' under 'Documentation and Downloads'.

4. Click on either the Standard Services tab or the Advanced Services tab on the WSDL Downloads component to find the functionality you are interested in.

5. Use the 'Download code in...' drop-down to select a programming language for the sample code of the WSDL that has the functionality you need and check the 'Include Documentation (PDF)' check-box under the drop-down in order to download the guide for that WSDL. Then select the gray 'Go' button beside the programming language you selected on the drop-down.

6. In the light gray margin near the name of the functionality (e.g. Ship Service, Rate Service, etc.), there is a link for the WSDL. Click on that link to download the desired WSDL.

7. If you wish to download all of the WSDLs' sample code, sample transactions, and the Developer Guide, at the top of the WSDL component, click the programming language of your choice on the drop-down next to 'DOWNLOAD ALL' and check the 'Include Documentation (PDF)' check-box. Then select the purple 'Download' button.

### 8.8.4 Notification

Table 58. Notification
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply/HighestSeverity/Notification</td>
<td>Includes the descriptive data detailing the status of a submitted transaction. Includes the severity of the notification, which indicates success or failure or some other information about the request. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>• SUCCESS – Your transaction succeeded with no other applicable information.</td>
</tr>
<tr>
<td></td>
<td>• NOTE – Additional information that may be of interest to you about your transaction.</td>
</tr>
<tr>
<td></td>
<td>• WARNING – Additional information that you need to know about your transaction that you may need to take action on.</td>
</tr>
<tr>
<td></td>
<td>• ERROR – Information about an error that occurred while processing your transaction.</td>
</tr>
<tr>
<td></td>
<td>• FAILURE – FedEx was unable to process your transaction.</td>
</tr>
<tr>
<td>Elements</td>
<td>Complex types</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>AddressValidationReply</td>
<td>Address</td>
</tr>
<tr>
<td>AddressValidationRequest</td>
<td>AddressAttribute</td>
</tr>
<tr>
<td></td>
<td>AddressToValidate</td>
</tr>
<tr>
<td></td>
<td>AddressValidationReply</td>
</tr>
<tr>
<td></td>
<td>AddressValidationRequest</td>
</tr>
<tr>
<td></td>
<td>AddressValidationResult</td>
</tr>
<tr>
<td></td>
<td>ClientDetail</td>
</tr>
<tr>
<td></td>
<td>Contact</td>
</tr>
<tr>
<td></td>
<td>Localization</td>
</tr>
<tr>
<td></td>
<td>Notification</td>
</tr>
<tr>
<td></td>
<td>NotificationParameter</td>
</tr>
<tr>
<td></td>
<td>ParsedAddressPartsDetail</td>
</tr>
<tr>
<td></td>
<td>ParsedPostalCodeDetail</td>
</tr>
<tr>
<td></td>
<td>ParsedStreetLineDetail</td>
</tr>
<tr>
<td></td>
<td>TransactionDetail</td>
</tr>
<tr>
<td></td>
<td>VersionId</td>
</tr>
<tr>
<td></td>
<td>WebAuthenticationCredential</td>
</tr>
<tr>
<td></td>
<td>WebAuthenticationDetail</td>
</tr>
</tbody>
</table>

**element AddressValidationReply**

- **diagram**: `AddressValidationReply`
- **namespace**: `http://fedex.com/ws/addressvalidation/v4`
- **type**: `ns:AddressValidationReply`
- **source**: `<xs:element name="AddressValidationReply" type="ns:AddressValidationReply"/>

**element AddressValidationRequest**

- **diagram**: `AddressValidationRequest`
- **namespace**: `http://fedex.com/ws/addressvalidation/v4`
- **type**: `ns:AddressValidationRequest`
- **source**: `<xs:element name="AddressValidationRequest" type="ns:AddressValidationRequest"/>

**complexType Address**
Descriptive data for a physical location. May be used as an actual physical address (place to which one could go), or as a container of "address parts" which should be handled as a unit (such as a city-state-ZIP combination within the US).
<xs:complexType name="Address">
  <xs:annotation>
    <xs:documentation>Descriptive data for a physical location. May be used as an actual physical address (place to which one could go), or as a container of "address parts" which should be handled as a unit (such as a city-state-ZIP combination within the US). </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="StreetLines" type="xs:string" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Combination of number, street name, etc. At least one line is required for a valid physical address; empty lines should not be included. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="City" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Name of city, town, etc. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="StateOrProvinceCode" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifying abbreviation for US state, Canada province, etc. Format and presence of this field will vary, depending on country. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PostalCode" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identification of a region (usually small) for mail/package delivery. Format and presence of this field will vary, depending on country. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="UrbanizationCode" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Relevant only to addresses in Puerto Rico. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="CountryCode" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The two-letter code used to identify a country. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="CountryName" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The fully spelt out name of a country. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Residential" type="xs:boolean" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Indicates whether this address residential (as opposed to commercial) </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
complexType AddressAttribute

documentation

Specifies additional information about the address processed by the SHARE systems as a key-value pair.

source

<xs:complexType name="AddressAttribute">
    <xs:annotation>
        <xs:documentation>Specifies additional information about the address processed by the SHARE systems as a key-value pair.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Name" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Specifies the key for the address attribute.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Value" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>The value for the key for the address attribute.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>

complexType AddressToValidate

documentation


complexType AddressToValidate

namespace http://fedex.com/ws/addressvalidation/v4
children ns1:ClientReferenceId ns1:Contact ns1:Address

source
<xs:complexType name="AddressToValidate">
  <xs:sequence>
    <xs:element name="ClientReferenceId" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>A reference id provided by the client.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Contact" type="ns:Contact" minOccurs="0"/>
    <xs:element name="Address" type="ns:Address" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

complexType AddressValidationReply

namespace http://fedex.com/ws/addressvalidation/v4
children ns1:HighestSeverity ns1:Notifications ns1:TransactionDetail ns1:Version ns1:ReplyTimestamp

source
<xs:complexType name="AddressValidationReply">
  <xs:sequence>
    <xs:element name="HighestSeverity" type="ns:NotificationSeverityType"/>
    <xs:element name="Notifications" type="ns:Notification" maxOccurs="1..∞"/>
    <xs:element name="TransactionDetail" type="ns:TransactionDetail"/>
    <xs:element name="Version" type="ns:Version"/>
    <xs:element name="ReplyTimestamp" type="xs:dateTime"/>
  </xs:sequence>
</xs:complexType>
complexType AddressValidationReply

<xs:complexType name="AddressValidationReply">
  <xs:sequence>
    <xs:element name="HighestSeverity" type="ns:NotificationSeverityType" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="Notifications" type="ns:Notification" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"/>
    <xs:element name="Version" type="ns:VersionId" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="ReplyTimestamp" type="xs:dateTime" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="AddressResults" type="ns:AddressValidationResult" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

complexType AddressValidationRequest

diagram

namespace http://fedex.com/ws/addressvalidation/v4

children ns1:WebAuthenticationDetail ns1:ClientDetail ns1:TransactionDetail ns1:Version
         ns1:InEffectAsOfTimestamp ns1:AddressesToValidate

source
<xs:complexType name="AddressValidationRequest">
  <xs:sequence>
    <xs:element name="WebAuthenticationDetail" type="ns:WebAuthenticationDetail" minOccurs="1">
      <xs:annotation>
        <xs:documentation>Descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services).</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="ClientDetail " type="ns:ClientDetail " minOccurs ="1"/>
<xs:element name="TransactionDetail " type="ns:TransactionDetail " minOccurs ="0"/>
<xs:element name="Version " type="ns:VersionId " minOccurs ="1"/>
<xs:element name="InEffectAsOfTimestamp " type="xs:dateTime " minOccurs ="0" maxOccurs ="unbounded "/>
<xs:element name="AddressesToValidate " type="ns:AddressToValidate " minOccurs ="0" maxOccurs ="unbounded "/>
</xs:sequence >
</xs:complexType >
<xs:element name="State" type="ns:OperationalAddressStateType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Specifies the degree to which SHARE service was able to cannonicalise the address provided, as per USPS standards and match it to an address already in the internal FedEx address repository.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Classification" type="ns:FedExAddressClassificationType" minOccurs="0"/>
<xs:element name="EffectiveContact" type="ns:Contact" minOccurs="0"/>
<xs:element name="EffectiveAddress" type="ns:Address" minOccurs="0"/>
<xs:element name="ParsedAddressPartsDetail" type="ns:ParsedAddressPartsDetail" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Additional attributes about the validated address from FedEx address repository.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:sequence>
</xs:sequence>
</xs:complexType>
**namespace**
http://fedex.com/ws/addressvalidation/v4

**children**
- ns1:AccountNumber
- ns1:MeterNumber
- ns1:IntegratorId
- ns1:Localization

**annotation**
Descriptive data for the client submitting a transaction.

**source**
```xml
<xs:complexType name="ClientDetail">
    <xs:annotation>
        <xs:documentation>Descriptive data for the client submitting a transaction.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="AccountNumber" type="xs:string" minOccurs="1">
            <xs:annotation>
                <xs:documentation>The FedEx account number associated with this transaction.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="MeterNumber" type="xs:string" minOccurs="1">
            <xs:annotation>
                <xs:documentation>This number is assigned by FedEx and identifies the unique device from which the request is originating.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```
<xs:element name="IntegratorId" type="xs:string" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Only used in transactions which require identification of the FedEx Office integrator.</xs:documentation>
    </xs:annotation>
</xs:element>

<xs:element name="Localization" type="ns:Localization" minOccurs="0">
    <xs:annotation>
        <xs:documentation>The language to be used for human-readable Notification.localizedMessages in responses to the request containing this ClientDetail object. Different requests from the same client may contain different Localization data. (Contrast with TransactionDetail.localization, which governs data payload language/translation.)</xs:documentation>
    </xs:annotation>
</xs:element>
namespace http://fedex.com/ws/addressvalidation/v4

children
ns1:ContactId  ns1:PersonName  ns1:Title  ns1:CompanyName  ns1:PhoneNumber
ns1:PhoneExtension  ns1:TollFreePhoneNumber  ns1:PagerNumber  ns1:FaxNumber
ns1:EMailAddress

annotation documentation
The descriptive data for a point-of-contact person.

```xml
<xs:complexType name="Contact">
  <xs:annotation>
    <xs:documentation>The descriptive data for a point-of-contact person.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ContactId" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Client provided identifier corresponding to this contact information.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PersonName" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the contact person's name.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Title" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the contact person's title.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="CompanyName" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the company this contact is associated with.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PhoneNumber" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the phone number associated with this contact.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PhoneExtension" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the phone extension associated with this contact.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TollFreePhoneNumber" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies a toll free number, if any, associated with this contact.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PagerNumber" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the pager number associated with this contact.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```
<xs:element name="Contact " type="xs:string " minOccurs="0">
  <xs:annotation>
    <xs:documentation>Identifies the contact information associated with this</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="FaxNumber " type="xs:string " minOccurs="0">
  <xs:annotation>
    <xs:documentation>Identifies the fax number associated with this contact. </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="EmailAddress " type="xs:string " minOccurs="0">
  <xs:annotation>
    <xs:documentation>Identifies the email address associated with this contact. </xs:documentation>
  </xs:annotation>
</xs:element>

complexType Localization
diagram

namespace http://fedex.com/ws/addressvalidation/v4
children ns1:LanguageCode ns1:LocaleCode
annotation documentation Identifies the representation of human-readable text.
source
<xs:complexType name="Localization ">
  <xs:annotation>
    <xs:documentation>Identifies the representation of human-readable text. </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="LanguageCode " type="xs:string " minOccurs="1">
      <xs:annotation>
        <xs:documentation>Two-letter code for language (e.g. EN, FR, etc.) </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="LocaleCode " type="xs:string " minOccurs="0">
      <xs:annotation>
        <xs:documentation>Two-letter code for the region (e.g. us, ca, etc.). </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
complexType Notification
diagram
The descriptive data regarding the result of the submitted transaction.

```xml
<xs:complexType name="Notification">
  <xs:sequence>
    <xs:element name="Severity" type="ns1:NotificationSeverityType" />
    <xs:element name="Source" type="xs:string" />
    <xs:element name="Code" type="xs:string" />
    <xs:element name="Message" type="xs:string" />
    <xs:element name="LocalizedMessage" type="xs:string" />
    <xs:element name="MessageParameters" type="ns1:NotificationParameter" maxOccurs="0..∞" />
  </xs:sequence>
</xs:complexType>
```
The descriptive data regarding the result of the submitted transaction.

**Severity**

The severity of this notification. This can indicate success or failure or some other information about the request. The values that can be returned are SUCCESS - Your transaction succeeded with no other applicable information. NOTE - Additional information that may be of interest to you about your transaction. WARNING - Additional information that you need to know about your transaction that you may need to take action on. ERROR - Information about an error that occurred while processing your transaction. FAILURE - FedEx was unable to process your transaction at this time due to a system failure. Please try again later.

**Source**

Indicates the source of this notification. Combined with the Code it uniquely identifies this notification.

**Code**

A code that represents this notification. Combined with the Source it uniquely identifies this notification.

**Message**

Human-readable text that explains this notification.

**LocalizedMessage**

The translated message. The language and locale specified in the ClientDetail. Localization are used to determine the representation. Currently only supported in a TrackReply.

**MessageParameters**

A collection of name/value pairs that provide specific data to help the client determine the nature of an error (or warning, etc.) without having to parse the message string.
complexType **NotificationParameter**

```xml
<xs:complexType name="NotificationParameter">
  <xs:sequence>
    <xs:element name="Id" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the type of data contained in Value (e.g. SERVICE_TYPE, PACKAGE_SEQUENCE, etc.).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Value" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The value of the parameter (e.g. PRIORITY_OVERNIGHT, 2, etc..).</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

**Namespace**: `http://fedex.com/ws/addressvalidation/v4`

**Children**: `ns1:Id` `ns1:Value`

**Source**:
```xml
<xs:complexType name="NotificationParameter">
  <xs:sequence>
    <xs:element name="Id" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the type of data contained in Value (e.g. SERVICE_TYPE, PACKAGE_SEQUENCE, etc.).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Value" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The value of the parameter (e.g. PRIORITY_OVERNIGHT, 2, etc..)</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

**complexType** **ParsedAddressPartsDetail**

```xml
<xs:complexType name="ParsedAddressPartsDetail">
  <xs:sequence>
    <xs:element name="ParsedPostalCode" type="ns:ParsedPostalCodeDetail"/>
    <xs:element name="ParsedStreetLine" type="ns:ParsedStreetLineDetail"/>
  </xs:sequence>
</xs:complexType>
```

**Namespace**: `http://fedex.com/ws/addressvalidation/v4`

**Diagram**

[Diagram of ParsedAddressPartsDetail]
### ParsedPostalCodeDetail

**Diagram**

```
ParsedPostalCodeDetail
```

**Source**

```xml
<xs:complexType name="ParsedPostalCodeDetail">
  <xs:sequence>
    <xs:element name="Base" type="xs:string" minOccurs="0"/>
    <xs:element name="AddOn" type="xs:string" minOccurs="0"/>
    <xs:element name="DeliveryPoint" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**Namespace**

http://fedex.com/ws/addressvalidation/v4

### ParsedStreetLineDetail

**Diagram**

```
```

**Source**

```xml
<xs:complexType name="ParsedStreetLineDetail">
  <xs:sequence>
    <xs:element name="Base" type="xs:string" minOccurs="0"/>
    <xs:element name="AddOn" type="xs:string" minOccurs="0"/>
    <xs:element name="DeliveryPoint" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**Namespace**

http://fedex.com/ws/addressvalidation/v4

### children

<table>
<thead>
<tr>
<th>ns1:ParsedStreetLine</th>
<th>ns1:ParsedPostalCode</th>
</tr>
</thead>
</table>

### source

```xml
<xs:complexType name="ParsedAddressPartsDetail">
  <xs:sequence>
    <xs:element name="ParsedStreetLine" type="ns:ParsedStreetLineDetail" minOccurs="0"/>
    <xs:element name="ParsedPostalCode" type="ns:ParsedPostalCodeDetail" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The postal code specified in a form that is supported by USPS as base, secondary and tertiary.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```
namespace http://fedex.com/ws/addressvalidation/v4

children ns1:HouseNumber  ns1:PreStreetType  ns1:LeadingDirectional  ns1:StreetName  ns1:StreetSuffix  ns1:TrailingDirectional  ns1:UnitLabel  ns1:UnitNumber  ns1:RuralRoute  ns1:POBox  ns1:Building  ns1:Organization

source
<xs:complexType name="ParsedStreetLineDetail">
  <xs:sequence>
    <xs:element name="HouseNumber" type="xs:string" minOccurs="0"/>
    <xs:element name="PreStreetType" type="xs:string" minOccurs="0"/>
    <xs:element name="LeadingDirectional" type="xs:string" minOccurs="0"/>
    <xs:element name="StreetName" type="xs:string" minOccurs="0"/>
    <xs:element name="StreetSuffix" type="xs:string" minOccurs="0"/>
    <xs:element name="TrailingDirectional" type="xs:string" minOccurs="0"/>
    <xs:element name="UnitLabel" type="xs:string" minOccurs="0"/>
    <xs:element name="UnitNumber" type="xs:string" minOccurs="0"/>
    <xs:element name="RuralRoute" type="xs:string" minOccurs="0"/>
    <xs:element name="POBox" type="xs:string" minOccurs="0"/>
    <xs:element name="Building" type="xs:string" minOccurs="0"/>
    <xs:element name="Organization" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="POBox" type="xs:string" minOccurs="0"/>
<xs:element name="Building" type="xs:string" minOccurs="0"/>
<xs:element name="Organization" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

complexType TransactionDetail
diagram

namespace http://fedex.com/ws/addressvalidation/v4
children ns1:CustomerTransactionId   ns1:Localization
source
<xs:complexType name="TransactionDetail">
  <xs:sequence>
    <xs:element name="CustomerTransactionId" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Free form text to be echoed back in the reply. Used to match requests and replies.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Localization" type="ns:Localization" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Governs data payload language/translations (contrasted with ClientDetail.localization, which governs Notification.localizedMessage language selection).</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

complexType VersionId
diagram
namespace http://fedex.com/ws/addressvalidation/v4

children ns1:ServiceId  ns1:Major  ns1:Intermediate  ns1:Minor

annotation documentation
Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).

source
<xs:complexType name="VersionId ">
  <xs:annotation>
    <xs:documentation>
      Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ServiceId" type="xs:string" fixed="aval" minOccurs="1">
      <xs:annotation>
        <xs:documentation>
          Identifies a system or sub-system which performs an operation.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Major" type="xs:int" fixed="4" minOccurs="1">
      <xs:annotation>
        <xs:documentation>
          Identifies the service business level.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Intermediate" type="xs:int" fixed="0" minOccurs="1">
      <xs:annotation>
        <xs:documentation>
          Identifies the service interface level.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="Minor" type="xs:int" fixed="0" minOccurs="1">
  <xs:annotation>
    <xs:documentation>Identifies the service code level.</xs:documentation>
  </xs:annotation>
</xs:element>

complexType **WebAuthenticationCredential**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram of WebAuthenticationCredential]</td>
</tr>
</tbody>
</table>

namespace http://fedex.com/ws/addressvalidation/v4

children ns1:Key ns1:Password

annotation documentation Two part authentication string used for the sender's identity

source

```xml
<xs:complexType name="WebAuthenticationCredential">
  <xs:annotation>
    <xs:documentation>Two part authentication string used for the sender's identity</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Key" type="xs:string" minOccurs="1">
      <xs:annotation>
        <xs:documentation>Identifying part of authentication credential. This value is provided by FedEx after registration</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Password" type="xs:string" minOccurs="1">
      <xs:annotation>
        <xs:documentation>Secret part of authentication key. This value is provided by FedEx after registration.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **WebAuthenticationDetail**

source

```xml
<xs:complexType name="WebAuthenticationDetail">
  <xs:annotation>
    <xs:documentation></xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Key" type="xs:string" minOccurs="1">
      <xs:annotation>
        <xs:documentation></xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Password" type="xs:string" minOccurs="1">
      <xs:annotation>
        <xs:documentation></xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```
namespace http://fedex.com/ws/addressvalidation/v4

children ns1:ParentCredential  ns1:UserCredential

annotation
Used in authentication of the sender’s identity.

source
<xs:complexType name="WebAuthenticationDetail ">
  <xs:annotation>
    <xs:documentation>Used in authentication of the sender’s identity.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ParentCredential " type="ns:WebAuthenticationCredential " minOccurs="0">
      <xs:annotation>
        <xs:documentation>This was renamed from cspCredential.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="UserCredential " type="ns:WebAuthenticationCredential " minOccurs="1">
      <xs:annotation>
        <xs:documentation>Credential used to authenticate a specific software application. This value is provided by FedEx after registration.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

---

simpleType AutoConfigurationType

namespace http://fedex.com/ws/addressvalidation/v4

type restriction of xs:string

properties base xs:string

facets Kind Value Annotation
enum EnterPRISE
enum SHIPPING_SERVICE_PROVIDER
enum SOFTWARE_ONLY
enum TRADITIONAL

source
<xs:simpleType name="AutoConfigurationType ">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ENTERPRISE "/>
  </xs:restriction>
</xs:simpleType>
simpleType FedExAddressClassificationType

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace</td>
<td>FedExAddressClassificationType</td>
</tr>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>facets</td>
<td>Kind Value Annotation</td>
</tr>
<tr>
<td></td>
<td>enumeration BUSINESS</td>
</tr>
<tr>
<td></td>
<td>enumeration MIXED</td>
</tr>
<tr>
<td></td>
<td>enumeration RESIDENTIAL</td>
</tr>
<tr>
<td></td>
<td>enumeration UNKNOWN</td>
</tr>
<tr>
<td>annotation</td>
<td>Specifies the address classification (business vs. residential)</td>
</tr>
<tr>
<td>source</td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:documentation</a>Specifies the address classification (business vs. residential)</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:documentation&gt;</td>
</tr>
</tbody>
</table>

simpleType NotificationSeverityType

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace</td>
<td>FedExAddressClassificationType</td>
</tr>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>facets</td>
<td>Kind Value Annotation</td>
</tr>
<tr>
<td></td>
<td>enumeration ERROR</td>
</tr>
<tr>
<td></td>
<td>enumeration FAILURE</td>
</tr>
<tr>
<td></td>
<td>enumeration NOTE</td>
</tr>
<tr>
<td></td>
<td>enumeration SUCCESS</td>
</tr>
<tr>
<td></td>
<td>enumeration WARNING</td>
</tr>
<tr>
<td>source</td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;ERROR&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;FAILURE&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;NOTE&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;SUCCESS&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;WARNING&quot;/&gt;</td>
</tr>
</tbody>
</table>
<xs:enumeration value="FAILURE ">
<xs:enumeration value="NOTE ">
<xs:enumeration value="SUCCESS ">
<xs:enumeration value="WARNING ">
</xs:restriction >
</xs:simpleType >

**simpleType** OperationalAddressStateType

<table>
<thead>
<tr>
<th>namespace</th>
<th><a href="http://fedex.com/ws/addressvalidation/v4">http://fedex.com/ws/addressvalidation/v4</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>base xs:string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
<th>Kind</th>
<th>Value</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>enumeration</td>
<td>NORMALIZED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enumeration</td>
<td>RAW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enumeration</td>
<td>STANDARDIZED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>annotation</th>
<th>documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specifies how different the address returned is from the address provided. This difference can be because the address is cannonialised to match the address specification standard set by USPS.</td>
</tr>
</tbody>
</table>

| source          | <xs:simpleType name="OperationalAddressStateType ">
|-----------------|<xs:annotation >
|                 | <xs:documentation >Specifies how different the address returned is from the address provided. This difference can be because the address is cannonialised to match the address specification standard set by USPS. </xs:documentation >
|                 | </xs:annotation >
|                 | <xs:restriction base="xs:string ">
|                 | <xs:enumeration value="NORMALIZED ">
|                 | <xs:enumeration value="RAW"/> 
|                 | <xs:enumeration value="STANDARDIZED "/>
|                 | </xs:restriction > 
|                 | </xs:simpleType >