FedEx.

FedEx Web Services

Close Service

August 2010

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 guarantees. FedEx may suspend any applicable money-backguarantees 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 under 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, the FedEx Freight 100-Series Rules Tariff, and the FedEx Service Guide shall be governed by the FedEx Automation Agreement and the FedEx Service Guide, in that order.

Contents

About This Guide

This guide describes how to work 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

The following may also be useful for FedEx Web Services developers:

- FedEx Services At-a-Glance: **fedex.com/us/services/ataglance.html**
- FedEx Service Guide: **fedex.com/us/services/pdf**
- Web Services organization home page: webservices.org
- Microsoft Web Services: msdn.microsoft.com/webservices
- O'Reilly XML.com: webservices.xml.com

Support

For FedEx Web Services technical support, you can reach FedEx at websupport@fedex.com or call 1.877.339.2774 and state “FedEx Web Services” at the voice prompt. Support hours are Monday through Friday, 7 a.m. to 9 p.m. (CST) and Saturday, 9 a.m. to 3 p.m. (CST). For international customer support, call 1.800.GoFedEx (1.800.463.3339).

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:

- Verify Addresses and Improve Customer Satisfaction: Prompt customers for additional information in the event of an address discrepancy or missing information with the Address Validation Service.
- Give Customers More Options: Help customers learn about all the available shipping options and rates with Ship Service and Rate Services. You can also extend this service to your shopping cart and Website, allowing customers to access money-saving information firsthand.
- More Convenience: Use the Locator Service to find the FedEx pickup location nearest your customer. Or, send an e-mail 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.
- Estimated duties and taxes calculations are now available. Contact your FedEx account executive for more information.
- 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 E-Mail Labels. This service sends an e-mail 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.

Document Overview

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

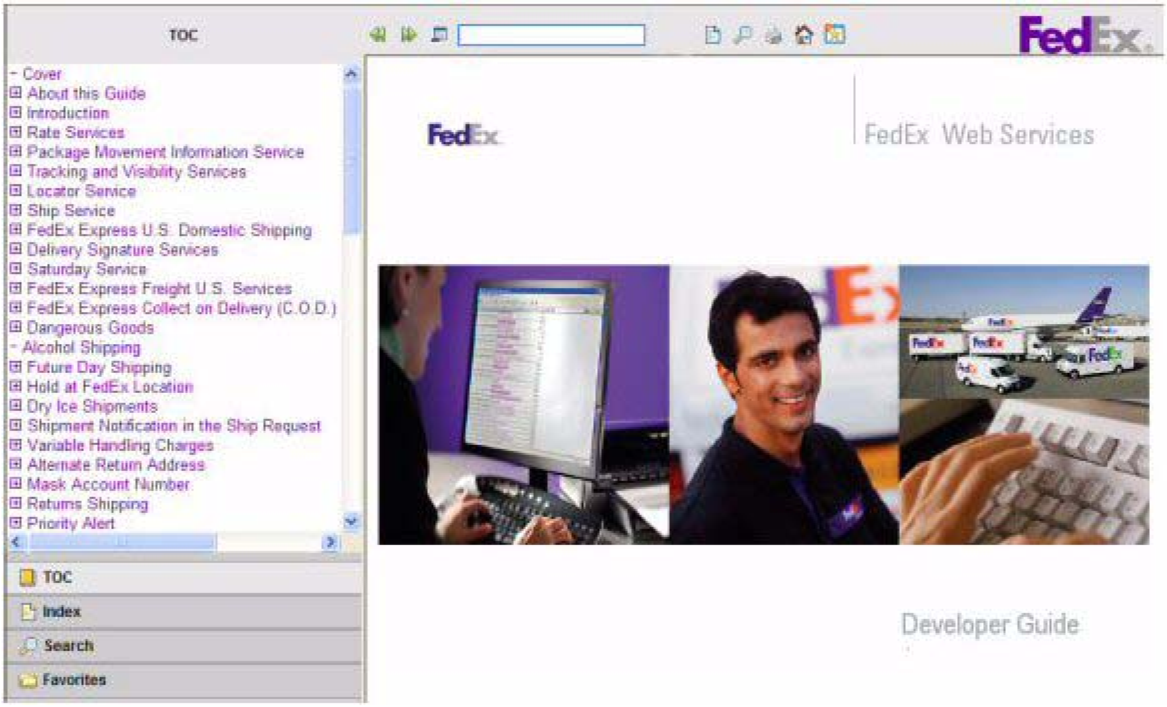
- Introduction (this chapter):
    - Documentation overview and guidelines, including how to use the Help application and how to print this guide.
    - Overview information about Web Services, including a high-level description of FedEx Web Services methods.
    - Coding basics.
    - Overview information about testing and certifying your application.
- Close Service allows you to reconcile shipping information for your ground shipments and print a ground manifest for your ground driver.

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.

Using the Web Services Online Help

This guide is available as online help at the FedEx Developer Resource Center (**fedex.com/developer**) in **Support** > **FedEx Web Services Developer Guide**. .



Web Services Help opens in your default browser, such as Internet Explorer or Firefox. The first topic—in this case, the cover page—appears in the main window. The Table of Contents (TOC) appears in the navigation column. Under the TOC you can choose the Index, Search, or Favorites options. Each of these features appears in the same column.

The toolbar across the top of the window displays the following elements:

Back: Returns you to the previously viewed topic.

Forward: Goes to the next topic as listed in the TOC.

Quick Search: Enter a search term and click to highlight the term in the current topic. This feature only searches the current topic.

Hide Navigation: Hides the left navigation column.

Search: Opens the full search tool in the navigation column.

Print: Opens the **Print** dialog box.

Home: Opens the default topic: in this case, the cover page.

Add Topic to Favorites: Saves the current topic to your **Favorites** list.

Printing This Guide or Online Help

You can print all or part of this guide from both the PDF and WebHelp versions.

Printing from the PDF Version

From the PDF version you can print the complete document or a page range of the document.

Open the PDF file and click the printer icon 🖨🖨 or click **File** > **Print**.

From the **Print** dialog box you can print the complete document, specify a page range, or choose from any of the available print options.

Printing from the WebHelp Version

From the WebHelp version you can print a single topic or a page range of that topic.

Open WebHelp and click the printer icon 🖨🖨.

From the **Print** dialog box you can print the complete topic or specify a page range.

Web Services, WSDL, and SOAP Overview

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

Web Services

Web Services is 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. .

Note that FedEx Web Services are not offered as part of a Universal Description Discovery and Integration (UDDI) and must be downloaded from the FedEx Developer Resource Center (**fedex.com/developer**) and stored locally for development and usage.

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>. .

Element	Definition
<definitions>	The root element contains name space definitions.
<portType>	The most important WSDL element. It is a set of all operations that a Web Service can accept and is a container for <operation> 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.
<types>	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).The <types> element defines the data types that are used by the Web Service For maximum platform neutrality, WSDL uses XML schema syntax to define data types.
<message>	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.
<operation>	Child of the <binding> element that defines each operation that the port exposes. This element allows three messages only: **Message** **Definition** Input Message Data Web Services receives Output Message Data Web Services sends Fault Message Error messages from Web Services
<service>	The <service> element contains a <port> child element that describes the URL where the service is located. This is the location of the ultimate Web Service.
<binding>	The <binding> element 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.

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

SOAP

SOAP is a simple XML-based protocol that allows applications to exchange information over HTTP. SOAP is built on open standards supported by numerous development tools on various platforms. SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. The SOAP request interface is an object in your application programming language.

SOAP enables the data to pass through layers of intermediaries and arrive at the ultimate receiver the way it was intended. It is worth noting that you may not need to actually construct the SOAP messages yourself—many development tools available today construct SOAP behind the scenes.

SOAP Message

A SOAP message is an ordinary 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:EnvelopeBody>

Example: Rate Request (SOAP Message)

```
<soapenv:Envelope xmlns:soapenv=”http://schemas.xmlsoap.org/soap/envelope/”
xmlns:v8=”http://fedex.com/ws/rate/v8”>
   <soapenv:Header/>
Optional    <SOAPsoapenv:HeaderBody>
      <v8:RateRequest>
          <v8:WebAuthenticationDetail>
            <v8:CspCredential>
               <v8:Key>Provider Key</v8:Key>
               <v8:Password>Provider Password</v8:Password>
            </v8:CspCredential>
            <v8:UserCredential>
               <v8:Key>User Key</v8:Key>
               <v8:Password>User Password</v8:Password>
            </v8:UserCredential>
         </v8:WebAuthenticationDetail>
         <v8:ClientDetail>
            <v8:AccountNumber>XXXXXXXXX</v8:AccountNumber>
            <v8:MeterNumber>XXXXXXX</v8:MeterNumber>
            <v8:ClientProductId>ABCD</v8:ClientProductId>
            <v8:ClientProductVersion>1234</v8:ClientProductVersion>
         </v8:ClientDetail>
         <v8:TransactionDetail>
            <v8:CustomerTransactionId>Test v8 Rate Request</
v8:CustomerTransactionId>
         </v8:TransactionDetail>
         <v8:Version>
            <v8:ServiceId>crs</v8:ServiceId>
            <v8:Major>8</v8:Major>
            <v8:Intermediate>0</v8:Intermediate>
            <v8:Minor>0</v8:Minor>
         </v8:Version>
<v8:ReturnTransitAndCommit>true</v8:ReturnTransitAndCommit>
         <v8:RequestedShipment>
            <v8:ShipTimestamp>2009-08-19T16:15:00-05:00</v8:ShipTimestamp>
            <v8:DropoffType>REGULAR_PICKUP</v8:DropoffType>
            <v8:PackagingType>YOUR_PACKAGING</v8:PackagingType>
            <v8:Shipper>
               <v8:Address>
                  <v8:StreetLines>123 Maple Street</v8:StreetLines>
                  <v8:City>Akron</v8:City>
                  <v8:StateOrProvinceCode>OH</v8:StateOrProvinceCode>
                  <v8:PostalCode>44333</v8:PostalCode>
                  <v8:CountryCode>US</v8:CountryCode>
               </v8:Address>
            </v8:Shipper>
            <v8:Recipient>
               <v8:Address>
                  <v8:StreetLines>123 Oak Street</v8:StreetLines>
                  <v8:City>Collierville</v8:City>
                  <v8:StateOrProvinceCode>TN</v8:StateOrProvinceCode>
                  <v8:PostalCode>38017</v8:PostalCode>
                  <v8:CountryCode>US</v8:CountryCode>
               </v8:Address>
            </v8:Recipient>
            <v8:RateRequestTypes>ACCOUNT</v8:RateRequestTypes>
            <v8:PackageCount>2</v8:PackageCount>
            <v8:PackageDetail>INDIVIDUAL_PACKAGES</v8:PackageDetail>
            <v8:RequestedPackages>
               <v8:SequenceNumber>1</v8:SequenceNumber>
               <v8:InsuredValue>
                  <v8:Currency>USD</v8:Currency>
                  <v8:Amount>50.00</v8:Amount>
               </v8:InsuredValue>
               <v8:Weight>
                  <v8:Units>LB</v8:Units>
                  <v8:Value>20.0</v8:Value>
               </v8:Weight>
               <v8:Dimensions>
                  <v8:Length>10</v8:Length>
                  <v8:Width>10</v8:Width>
                  <v8:Height>10</v8:Height>
                  <v8:Units>IN</v8:Units>
               </v8:Dimensions>
               <v8:CustomerReferences>
                  <v8:CustomerReferenceType>CUSTOMER_REFERENCE</
v8:CustomerReferenceType>
                  <v8:Value>TEST v8 REQUEST</v8:Value>
               </v8:CustomerReferences>
            </v8:RequestedPackages>
            <v8:RequestedPackages>
               <v8:SequenceNumber>2</v8:SequenceNumber>
               <v8:InsuredValue>
                  <v8:Currency>USD</v8:Currency>
                  <v8:Amount>50.00</v8:Amount>
               </v8:InsuredValue>
               <v8:Weight>
                  <v8:Units>LB</v8:Units>
                  <v8:Value>10.0</v8:Value>
               </v8:Weight>
               <v8:Dimensions>
                  <v8:Length>5</v8:Length>
                  <v8:Width>5</v8:Width>
                  <v8:Height>5</v8:Height>
                  <v8:Units>IN</v8:Units>
               </v8:Dimensions>
            </v8:RequestedPackages>
```

```
         </v8:RequestedShipment>
      </v8:RateRequest>
Required      <SOAP/soapenv:Body>
Required <SOAP/soapenv:BodyEnvelope>
```

Example: Delete Tag Request (SOAP Message)

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/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/v8">
<SOAP-ENV:Body>
<DeleteTagRequest>
<WebAuthenticationDetail>
<UserCredential>
<Key>xxxxxxxxxxxxxx</Key>
<Password></Password>
</UserCredential>
</WebAuthenticationDetail>
<ClientDetail>
<AccountNumber>xxxxxxxxx</AccountNumber>
<MeterNumber>xxxxxx</MeterNumber>
</ClientDetail>
<TransactionDetail>
<CustomerTransactionId>DE_Shakeout_wsvc</CustomerTransactionId>
</TransactionDetail>
<Version>
<ServiceId>ship</ServiceId>
<Major>8</Major>
<Intermediate>0</Intermediate>
<Minor>0</Minor>
</Version>
<DispatchLocationId>MQYA</DispatchLocationId>
<DispatchDate>2008-10-08</DispatchDate>
<Payment>
<PaymentType>shipper</PaymentType>
<Payor>
<AccountNumber>xxxxxxxxx</AccountNumber>
<CountryCode>US</CountryCode>
</Payor>
</Payment>
<ConfirmationNumber>997037200019454</ConfirmationNumber>
</DeleteTagRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

For more information about the SOAP standard, refer to the World Wide Web Consortium (W3C) Website at w3.org/TR/SOAP.

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 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. An example of a request using the non-SOAP interface looks like this:

```
<ns:TrackRequest xmlns:ns="http://fedex.com/ws/track/v5" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://fedex.com/ws/
track/v4 TrackService_v4.xsd ">
     <ns:WebAuthenticationDetail>
          <ns:UserCredential>
               <ns:Key>xxxxxxxxxxxxxx</ns:Key>
               <ns:Password>xxxxxxxxxxxxxxxxxxxxxxxx</ns:Password>
          </ns:UserCredential>
     </ns:WebAuthenticationDetail>
     <ns:ClientDetail>
          <ns:AccountNumber>000000000</ns:AccountNumber>
          <ns:MeterNumber>0000000</ns:MeterNumber>
     </ns:ClientDetail>
     <ns:TransactionDetail>
     <ns:CustomerTransactionId>User Customizable Field</
   ns:CustomerTransactionId></ns:TransactionDetail>
     <ns:Version>
          <ns:ServiceId>trck</ns:ServiceId>
          <ns:Major>4</ns:Major>
          <ns:Intermediate>0</ns:Intermediate>
          <ns:Minor>0</ns:Minor>
     </ns:Version>
     <ns:PackageIdentifier>
          <ns:Value>tttttttttttttt</ns:Value>
          <ns:Type>TRACKING_NUMBER_OR_DOORTAG</ns:Type>
     </ns:PackageIdentifier>
     <ns:IncludeDetailedScans>true</ns:IncludeDetailedScans>
</ns:TrackRequest>
```

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:

```
<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 xmlns:ship="http://fedex.com/ws/ship/
                  v8">000000000</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/8">
                     <ship:AccountNumberx>000000000000000000</
                     ship:AccountNumberx>
                     <ship:MeterNumber>0000000</ship:MeterNumber>
                  </ship:ClientDetail>
               </con1:xmlLocation>
            </con1:ValidationFailureDetail>
         </con:details>
         <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:fault>
   </detail>
</soapenv:Fault>
```

Each reply should be checked for the Fault element to indicate failure in processing the message. Note that the 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. For more information on Base64 decoding, see Creating a Label.

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:

```
POST /xml HTTP/1.0
Referer: YourCompanyNameGoesHere
Host: gatewaybetawsbeta.fedex.com
Port: 443
Accept: image/gif, image/jpeg, image/pjpeg, text/plain, text/html, */*
Content-Type: image/gif
Content-length: %d
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.

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.

Implementing FedEx Web Services

Before you begin your implementation of FedEx Web Services, make note of the following guidelines:

- FedEx Web Services is 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 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. .

- 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.

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. Notice that 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 associated with a FedEx Web Service. Other simple types are defined as part of XML schema's repertoire of built-in simple types. .
- XML schema built-in simple types are prefixed by “"xs:”", which is associated with the XML schema namespace through the declaration xmlns:xs="http://www.w3.org/2001//XMLSchema", displayed in the schema element.
- The same prefix, and the same association, are also part of the names of built-in simple types, e.g., . xs:string. This association identifies the elements and simple types as belonging to the vocabulary of the XML schema language, rather than the vocabulary of the schema author.

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.

XML Schema Diagrams

XML schema diagrams describe the elements (usually associated with a request or reply), complex types, and simple types that make up the WSDL. The following table illustrates the relationships and behavior of elements and types.

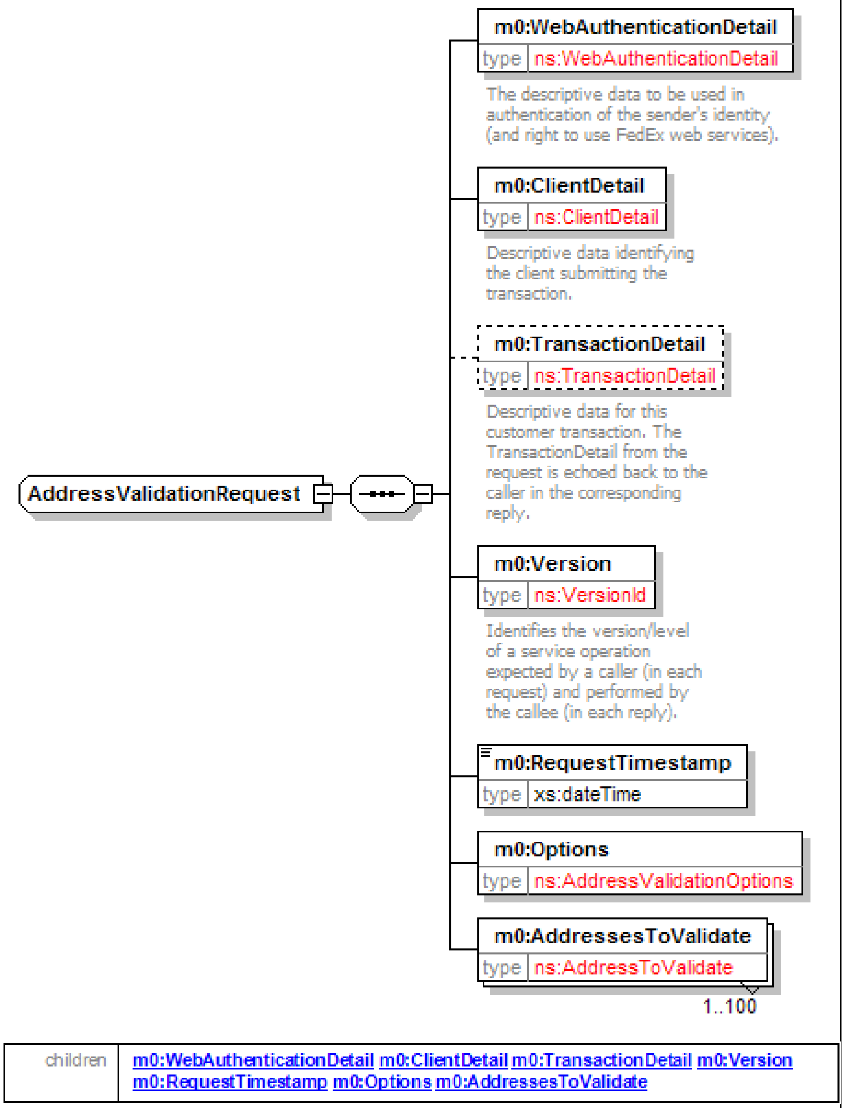
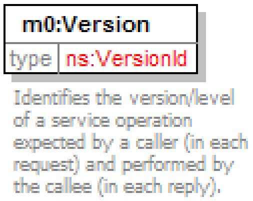
Diagram	Description
 AddressValidationRequest m0:WebAuthenticationDetail type ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). m0:ClientDetail type ns:ClientDetail Descriptive data identifying the client submitting the transaction. m0:TransactionDetail type ns:TransactionDetail Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply. m0:Version type ns:VersionId Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply). m0:RequestTimestamp type xs:dateTime m0:Options type ns:AddressValidationOptions m0:AddressesToValidate type ns:AddressToValidate 1..100 children m0:WebAuthenticationDetail m0:ClientDetail m0:TransactionDetail m0:Version m0:RequestTimestamp m0:Options m0:AddressesToValidate	Diagrams of a parent element, such as AddressValidationRequest, include connections to the child elements. Child elements can be simple or complex types. A child element connected with a solid line and surrounded by a box with a solid border represents a required type, such as ClientDetail. A child element connected by a dotted line and surrounded by a dotted border represents an optional type (minOccurs="0"), such as TransactionDetail. **Note:** An element that is defined as minOccurs="0" may be required for some calls. Types that are documented include the documentation directly below the box. All children are linked by name below the diagram.
 m0:Version type ns:VersionId Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).	A box with a single solid border represents a single element that is required. The type can be simple or complex.

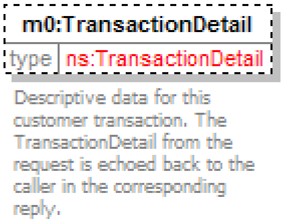
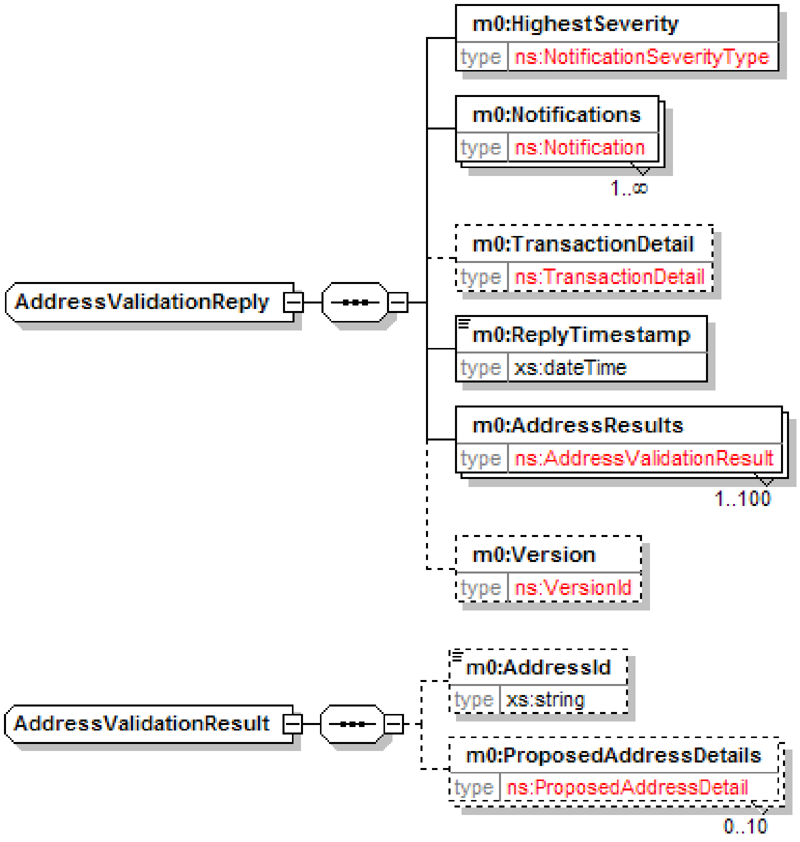
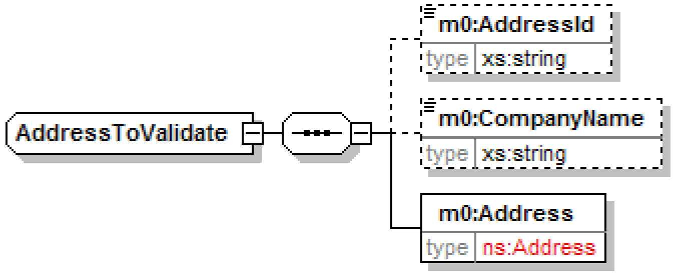
Diagram	Description
 m0:TransactionDetail type ns:TransactionDetail Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.	A box with a dotted border indicates a single element that is optional. The type can be simple or complex.
 AddressValidationReply m0:HighestSeverity type ns:NotificationSeverityType m0:Notifications type ns:Notification 1..∞ m0:TransactionDetail type ns:TransactionDetail m0:ReplyTimestamp type xs:dateTime m0:AddressResults type ns:AddressValidationResult 1..100 m0:Version type ns:VersionId AddressValidationResult m0:AddressId type xs:string m0:ProposedAddressDetails type ns:ProposedAddressDetail 0..10	A layered box represents a multiple occurrence element. A solid line represents a required multiple occurrence element. The number of possible occurrences appears below the box, as depicted by the AddressResults element. An unbounded number of occurrences is represented by the infinity ∞ symbol (maxOccurs="unbounded"), as depicted by the Notifications type. A layered box with a dotted border represents an optional multiple occurrence type (minOccurs="0"), such as ProposedAddressResults. **Note:** An element that is defined as minOccurs="0" may be required for some calls.
 AddressToValidate m0:AddressId type xs:string m0:CompanyName type xs:string m0:Address type ns:Address	A standard type such as “string” appears in black text below element name. A FedEx-specific type such as “Address” appears in red text below the element name.

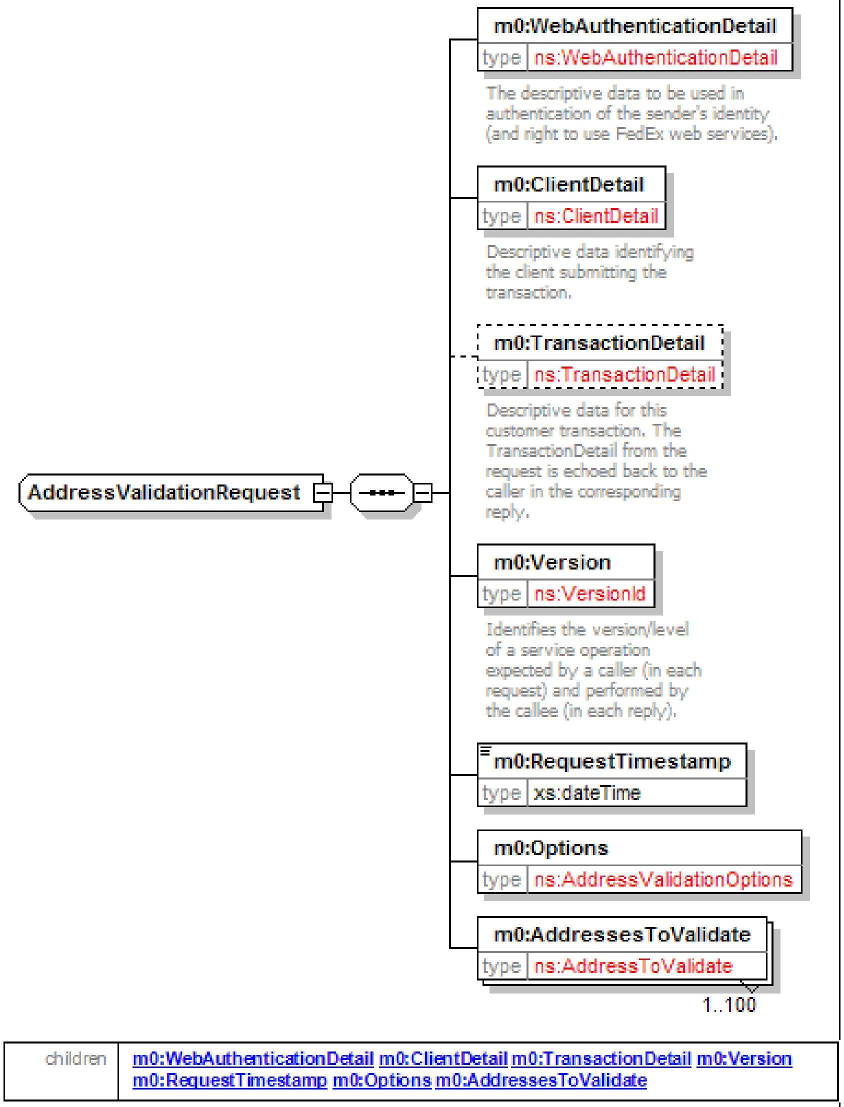
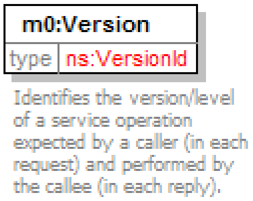
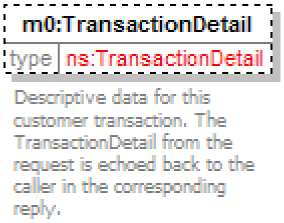
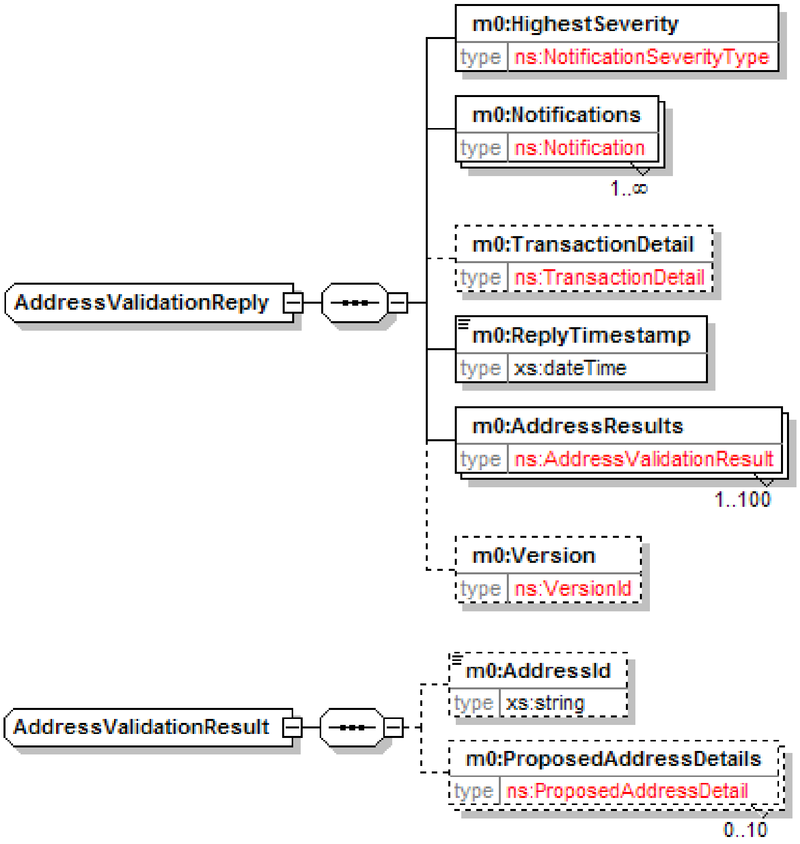
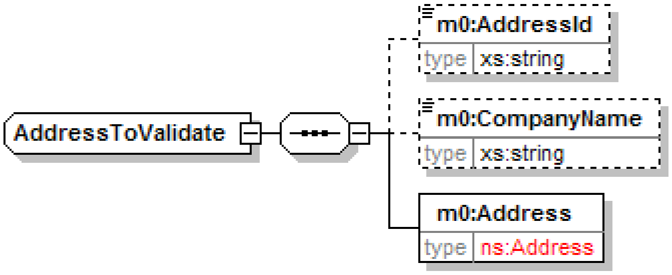
Diagram	Description
 AddressValidationRequest m0:WebAuthenticationDetail type ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). m0:ClientDetail type ns:ClientDetail Descriptive data identifying the client submitting the transaction. m0:TransactionDetail type ns:TransactionDetail Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply. m0:Version type ns:VersionId Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply). m0:RequestTimestamp type xs:dateTime m0:Options type ns:AddressValidationOptions m0:AddressesToValidate type ns:AddressToValidate 1..100 children m0:WebAuthenticationDetail m0:ClientDetail m0:TransactionDetail m0:Version m0:RequestTimestamp m0:Options m0:AddressesToValidate	Diagrams of a parent element, such as AddressValidationRequest, include connections to the child elements. Child elements can be simple or complex types. A child element connected with a solid line and surrounded by a box with a solid border represents a required type, such as ClientDetail. A child element connected by a dotted line and surrounded by a dotted border represents an optional type (minOccurs="0"), such as TransactionDetail. **Note:** An element that is defined as minOccurs="0" may be required for some calls. Types that are documented include the documentation directly below the box. All children are linked by name below the diagram.
 m0:Version type ns:VersionId Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).	A box with a single solid border represents a single element that is required. The type can be simple or complex.

Diagram	Description
 m0:TransactionDetail type ns:TransactionDetail Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.	A box with a dotted border indicates a single element that is optional. The type can be simple or complex.
 AddressValidationReply m0:HighestSeverity type ns:NotificationSeverityType m0:Notifications type ns:Notification 1..∞ m0:TransactionDetail type ns:TransactionDetail m0:ReplyTimestamp type xs:dateTime m0:AddressResults type ns:AddressValidationResult 1..100 m0:Version type ns:VersionId AddressValidationResult m0:AddressId type xs:string m0:ProposedAddressDetails type ns:ProposedAddressDetail 0..10	A layered box represents a multiple occurrence element. A solid line represents a required multiple occurrence element. The number of possible occurrences appears below the box, as depicted by the AddressResults element. An unbounded number of occurrences is represented by the infinity ∞ symbol (maxOccurs="unbounded"), as depicted by the Notifications type. A layered box with a dotted border represents an optional multiple occurrence type (minOccurs="0"), such as ProposedAddressResults. **Note:** An element that is defined as minOccurs="0" may be required for some calls.
 AddressToValidate m0:AddressId type xs:string m0:CompanyName type xs:string m0:Address type ns:Address	A standard type such as “string” appears in black text below element name. A FedEx specific type such as “Address” appears in red text below the element name.

Required Elements

Most requests to FedEx require the following complex elements:

Note: **T**hese elements are common to most Web Services (see the following table for which WSDLs need which common elements) and are not documented service by service.

WebAuthenticationDetail: The WebAuthenticationDetail element includes user credentials issued by FedEx so that your transactions are recognized by the FedEx back-end systems. The following elements are required:

Element	Description
WebAuthenticationDetail	The descriptive data to be used in authentication of the shipper's identity and right to use FedEx Web Services.
UserCredential	Credential used to authenticate a specific software application. This value is provided by FedEx after registration.
WebAuthenticationCredential	Two-part authentication string used to verify shipper identity.
WebAuthenticationCredential/Key	Unique identifier assigned to each customer as part of their authentication credentials.
WebAuthenticationCredential/Password	Second part of the authentication credential which has a direct relationship with the credential key.

Note: Web Services now use two-factor authentication. If you do not have new credentials, the latest WSDLs will use your old authentication credentials. If you do not have a new user authentication credential, do not populate the password element.

ClientDetail: The ClientDetail element is required for all services that need your FedEx account number and meter number. Requirements are:

Element	Description
ClientDetail/AccountNumber	Required. Your FedEx account number.
ClientDetail/MeterNumber	Maximum of 9 characters. The associated meter number for your FedEx account number.

- A standard type such as “string” appears in black text below element name.
- A FedEx specific type such as “Address” appears in red text below the element name.

Note: When building a Web-based application for shipping that will be used at multiple locations, include the local FedEx Express® account and meter in the ClientDetail section of the ship transaction. Create a database to hold multiple account and meter numbers for the shipping locations.

TransactionDetail: The TransactionDetail element is optional for all transactions. However, if you want to identify associated request and reply transactions, use this element.

Element	Description
TransactionDetail/CustomerTransactionId	Maximum of 40 characters. This element allows you to assign a unique identifier to your transaction. This element is returned in the reply and helps you match requests to replies.

VersionId: The VersionId element is required and uploads the WSDL version number to FedEx. FedEx provides the latest version number for the service you are using. This number should be updated when you implement a new version of the service.

Element	Description
ServiceId	Identifies a system or sub-system which performs an operation.
Major	Identifies the service business level.
Intermediate	Identifies the service interface level.
Minor	Identifies the service code level.

Shipper Information: Your shipper information is required for all shipping transactions: :

Element	Description
AccountNumber	If you include this element in the ship request, this entry overrides the account number in the ClientDetail element.
TIN	Tax Identification Number— — this information is required for international shipments only.
Contact	The Contact element includes: • PersonName • Title • CompanyName • Department • PhoneNumber • PagerNumber • FaxNumber • e-mailAddressEmailAddress
Address	This element includes: • StreetLines: two StreetLines elements are allowed. • City • StateOrProvinceCode: required if your shipper address is in the U.S. or Canada. • PostalCode: required. • UrbanizationCode: may be required if your shipper address is in Puerto Rico. • CountryCode: required.
Residential	Required if your shipper address is considered a residential location. If you are unsure, use the Address Validation Service to check your address.

Reply Notifications

Notifications are returned in replies. The notification element provides the notification ranked according to their severity:

- HighestSeverity: This element ranks the level of notification severity. Values include:
- FAILURE: Code/message explains that your request could not be handled at this time; do not resubmit right now.

- ERROR: Code/message identifies a problem with your request data; you may fix the request data and try again.
- WARNING: Your request was successful. However, the code/message explains what had to be done to fulfill your request; you may need to determine whether that is what you intended, you may need to do this differently next time, or you may need to prepare for a future change. Request was completed.
- NOTE: Your request was successful. However, the code/message contains additional information about how your request was fulfilled; you do not need to take any special action.
- SUCCESS: Your request was successful. There are no NOTE or WARNING notifications.

Note: There is a possibility of multiple Notification objects (different severity levels) for a single request. The response notification severity values of ERROR, FAILURE, and SUCCESS severity should never be combined in a single response.

Notification Examples

For example, if you need to perform a U.S. address correction, the service should accept a (U.S.) Address object from its client and return that Address in a standardized form (canonical spelling and abbreviation of street name parts, elimination of redundant white space, data correction where possible, etc.). The following cases illustrate several notification types.

The example service has been assigned a NotificationSourceType value of "USACS".

Case	Reply
Request to submit an Address that is valid and is already in standardized form (i.e., there is nothing to say except "OK").	Notifications: SUCCESS and Address: the original address (or copy).
Request to submit an Address that is valid but not in standardized form (e.g., the word "Boulevard" in a street name is replaced with the standard abbreviation "Blvd" and "Saint Louis" as a city name is replaced with "St Louis").	Notifications: NOTE, "Standard abbreviation applied to street name"}, {NOTE, "Standard abbreviation applied to city name"} and Address: the original address, with modification made to the street name and city name.
Request to submit an Address that is valid but with only a 5-digit postal code: the service supplies the ZIP+4 for the standardized address.	Notifications: NOTE, "ZIP+4 suffix added"} and Address: the original address, with the four-digit suffix added to the postal code.
Request to submit an Address that is identifiable by street data, city name, and state code, but with a 5-digit postal code that does not match the other fields. The service supplies the correct ZIP+4 for the standardized address.	Notifications: WARNING, "ZIP code corrected to match rest of address"} and Address: the original address, with the replacement postal code.

Case	Reply
Request to submit an Address that has an incorrect state code. The original address contains a ZIP+4 postal code belonging to a city/ state pair that matches the client's original city and street address. The service supplies the corresponding state code in the corrected address.	Notifications: WARNING, "State code corrected to match city and ZIP code"} and Address: the original address, with the revised state code.
Request to submit an Address that has an incorrect state code. The original address contains a ZIP+4 postal code belonging to a city/ state pair that matches the client's original city and street address. The service rejects the client's address.	Notifications: ERROR, "State code is incorrect for city/ZIP combination"} and Address: empty (either all fields blank or no Address at all).
Request to submit an Address that contains only a single street line (no city, state, or postal code). The service rejects the request.	Notifications: {ERROR, "City name is missing and cannot be corrected"}, {ERROR, "State code is missing and cannot be corrected"}, {ERROR, "ZIP code is missing and cannot be corrected"} and Address: empty (either all fields blank or no Address at all)
Request to submit an Address, but the address correction service's database server is down or fails.	Notifications: {FAILURE, "Service temporarily unavailable"}, Address: empty (either all fields blank or no Address at all).

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 conforms to industry standards and is compatible with a comprehensive array of developer’s 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.html.

To obtain FedEx Web Services and begin integrating with an application, you will need to access documentation, sample code, and sample service requests and replies with the WSDLs from the FedEx Developer Resource Center Technical Resources. You will also need to obtain a test meter number to engage in real-time online testing in the FedEx- hosted test environment. . Note that not all services are available outside the U.S.

Testing

FedEx supplies a complete online operating environment with which to test your applications against live FedEx servers. In order 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.

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.

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.

Once an application has completed the above mentioned process and requirements, FedEx will enable the provider's CSP credentials for processing all applicable services in the production environment. The URL needed to direct the CSP application to the FedEx production servers will also be provided. The provider would then need to obtain production User Credentials (Register CSP User Service) and a production meter number (Subscribe Service). Once this information has been obtained with the connection directed to the production servers, the provider's application is considered live.

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 you are planning to resell your solution to others.

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. .

Requirements and Resources for Corporate Developers	
Must be accepted into the FedEx® Compatible Solutions Program (CSP)	No
Self-certification of implementations using standard services	Yes
Self-certification of implementations using advanced services	No
Certification Assistance	Yes (WISC team)
FedEx supports the customer directly	Yes

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 through Friday, 7 a.m. to 9 p.m. (CST), 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 part of the FedEx Compatible Solutions Program. 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 CSP providers, go to the Available CSP Solutions page at **fedex.allegis.com/LeadReg.asp**.

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 to FedEx in order to obtain a production test meter.

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. .

Requirements and Resources for Commercial Developers	
Must be accepted into the FedEx Compatible Solutions Program (CSP)	Yes (Required)
Self-certification of implementation using Standard Services	No
Self-certification of implementations using Advanced Services	No
Certification Assistance	Yes (via CSP)
FedEx supports the customer directly	No
FedEx supports the commercial developer’s customer	Indirectly

If you are a commercial developer interested in becoming a FedEx Compatible Solutions Provider, visit **fedex.com/us/compatiblesolutions/provider/** for more information about the FedEx Compatible Solutions Program (CSP).

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.

Close Shipment

The Close Service WSDL allows you to reconcile shipping information for your FedEx Ground® or FedEx SmartPost® shipments and print a ground manifest for your ground driver. The ground manifest is generated after a successful close and must be printed before your ground shipments are tendered to FedEx. You may continue to ship ground packages after a close has been performed. Similarly, FedEx SmartPost must be closed in order for the system to become aware of the package before it is scanned at the SmartPost facility.

Ground Close Ship Day Service Details

The Close Service can be performed multiple times during your shipping day. Use the Close Service to initiate the final stage of processing all shipment data submitted by the customer for the day (close of business). In addition to the ground manifest, the following reports are returned after a successful close and need to be printed (if shipments with these services are included in the Close transaction):

- C.O.D./E.C.O.D.
- Hazmat
- Multiweight Package Detail

C.O.D./E.C.O.D. and Hazmat reports can be printed anytime during the shipping day for shipments that have already been closed (up to three days in the past). Multiweight reports (if the account is multiweight enabled) are produced at close time and cannot be reprinted.

Note: Close is a FedEx Ground®-only requirement. FedEx Express® shipments are automatically closed and do not require you to perform a specific close operation.

For more detailed information about the services offered by FedEx, see the electronic *FedEx Service Guide*.

Ground Close Ship Day Coding Details

The following elements are required for a Close Service transaction:

Element	Requirements
GroundCloseRequest	Initiates the final stage of processing all shipment data submitted by the customer for the day (close of business) for FedEx Ground shipments.
SmartPostCloseRequest	Initiates the final stage of processing all shipment data submitted by the customer for the day (close of business) for FedEx SmartPost shipments.
TimeUpToWhichShipmentsAreToBe Closed	Identifies the date and time up to which shipments are to be closed. Both date and time should be included in the Close request.

The following elements are required for a Ground Close Service with Documents request:

Element	Requirements
WebAuthenticationDetail	The descriptive data to be used in authentication of the sender's identity (and right to use FedEx Web Services).
ClientDetail	Descriptive data identifying the client submitting the transaction.
TransactionDetail	Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.
Version	Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).
CloseDate	Date on which shipments were closed.
CloseDocumentSpecification	Specifies characteristics of document(s) to be returned for this request. You may request the following close documents. • COD_REPORT • MANIFEST • MULTIWEIGHT_REPORT • OP_950

The following elements may be returned in the Ground Close with Documents reply:

Element	Requirements
HighestSeverity	Indicates the highest level of severity of all the notifications returned in this reply.
Notifications	Descriptive data regarding the results of the submitted transaction.
TransactionDetail	Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.
Version	Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).
Close Document Type	Actual document contents for all provided reports.

Ground Report Printing

When you perform a Close, Web Services generates four ground reports:

- Ground Manifest
- C.O.D./E.C.O.D. Report
- Multiweight Detail
- Hazardous Materials Certification Report (U.S. only)

You can also choose to reprint C.O.D./E.C.O.D., Ground Manifest and Hazmat reports using the Close service. Multiweight reports (if the account is multiweight enabled) are produced at close time and cannot be reprinted

Additional elements for printing ground reports include the following:

Element	Requirements
GroundCloseReportsReprintRequest	If this element is set to true, you receive ground reporting from the last three closed shipping days. To indicate which report(s) you want to receive, include the CloseReport Type element. and specify a date from the previous three calendar days. If this element is set to false, the Close Service processes and all valid reporting is returned.
CloseReportType	If the CloseRequest/ReportOnly is set to true, you may request the following reports: • MANIFEST • COD • HAZMAT • MULTIWEIGHT
TrackingNumber	If you include the tracking number of a particular FedEx Ground shipment for which you want to print ground reports (manifest, C.O.D., Hazmat, or Multiweight Detail), the report(s) requested in the CloseReportType element are returned.

SmartPost Close Shipment Service Details

The Close Service can be performed multiple times during your shipping day. Use the Close Service to initiate the final stage of processing all shipment data submitted by the customer for the day (close of business).

- SmartPost Close is independent of a FedEx Ground® Close. FedEx Express® shipments are automatically closed and do not require you to perform a specific close operation.
- No reports are returned in the SmartPost Close Reply.
- For more detailed information about the services offered by FedEx, see the electronic *FedEx Service Guide*.

SmartPost Close Ship Day Coding Details

The following elements are required for a Close Service transaction:

Element	Requirements
WebAuthenticationDetail/ UserCredential/Key	The value for the Key was provided by FedEx upon your request. It is an alphanumeric mixed-case string.
WebAuthenticationDetail/ UserCredential/Password	The value for the Password was provided by FedEx upon your request. It is an alphanumeric mixed-case string.
ClientDetail/AccountNumber	The FedEx Express nine-digit meter number that has been used in the Ship transactions

Element	Requirements
ClientDetail/MeterNumber	The FedEx Express nine-digit meter number that has been used in the Ship transactions.
Version/ServiceId	The value is "clos"
Version/Major	The current value is "2".
Version/Intermediate	The current value is "0".
Version/Minor	The current value is "0".
HubId	Specify the HubId. Valid values are: • 5303 ATGA Atlanta • 5281 CHNC Charlotte • 5602 CIIL Chicago • 5929 COCA Chino • 5751 DLTX Dallas • 5802 DNCO Denver • 5481 DTMI Detroit • 5087 EDNJ Edison • 5431 GCOH Grove City • 5771 HOTX Houston • 5465 ININ Indianapolis • 5648 KCKS Kansas City • 5902 LACA Los Angeles • 5254 MAWV Martinsburg • 5379 METN Memphis • 5552 MPMN Minneapolis • 5531 NBWI New Berlin • 5110 NENY Newburgh • 5015 NOMA Northborough • 5327 ORFL Orlando • 5194 PHPA Philadelphia • 5854 PHAZ Phoenix • 5150 PTPA Pittsburgh • 5958 SACA Sacramento • 5843 SCUT Salt Lake City • 5983 SEWA Seattle • 5631 STMO St. Louis **Note:** Include only the numeric HubId value in your request.
DestinationCountryCode	For SmartPost the value is always “US”
PickupCarrier	The value will be either “FDXG” or “FXSP”, depending on which driver picks up your SmartPost packages.

Element	Requirements
CustomerManifestId	Optional. Maximum length 8 characters. If no CustomerManifestId values were sent in the individual Ship request, then do not include this element in the request at all. If CustomerManifestId values were sent in the Ship requests, then send one SmartPost close for each unique CustomerManifestId. Each Close request will only close the packages with a matching CustomerManifestId value.

XML Schema

See Schema Close Service.

Schema CloseService_v2.xsd

targetNamespace: **http://fedex.com/ws/close/v2**

Elements

- GroundCloseDocumentsReply
- GroundCloseReply
- GroundCloseReportsReprintReply
- GroundCloseReportsReprintRequest
- GroundCloseRequest
- GroundCloseWithDocumentsRequest
- ReprintGroundCloseDocumentsRequest
- SmartPostCloseReply
- SmartPostCloseRequest

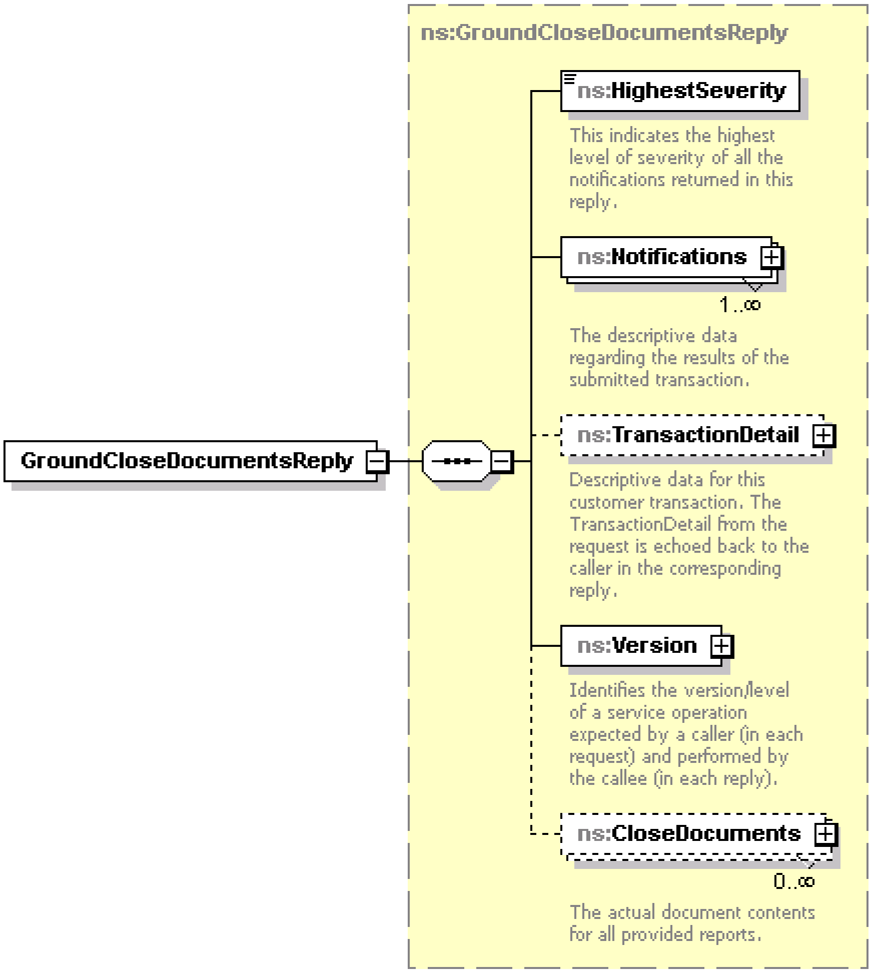
Complex types

- ClientDetail
- CloseDocument
- CloseDocumentFormat
- CloseDocumentSpecification
- CustomerImageUsage
- GroundCloseDocumentsReply
- GroundCloseReply
- GroundCloseReportsReprintReply
- GroundCloseReportsReprintRequest
- GroundCloseRequest
- GroundCloseWithDocumentsRequest
- LinearMeasure
- Localization
- ManifestFile
- Notification
- NotificationParameter
- Op950Detail
- ReprintGroundCloseDocumentsRequest
- ShippingDocumentDispositionDetail
- ShippingDocumentEMailDetail
- ShippingDocumentEMailRecipient
- ShippingDocumentPart
- ShippingDocumentPrintDetail
- ShippingDocumentStorageDetail
- SmartPostCloseReply
- SmartPostCloseRequest
- TransactionDetail
- VersionId
- WebAuthenticationCredential
- WebAuthenticationDetail

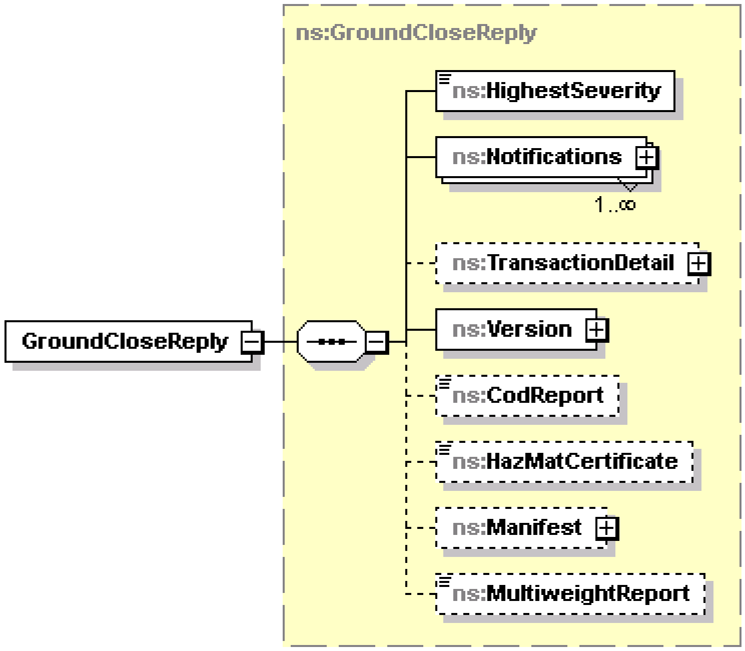
Simple types

- CarrierCodeType
- CloseDocumentType
- CloseReportType
- CustomerImageUsageType
- EMailNotificationRecipientType
- ImageId
- InternalImageType
- LinearUnits
- NotificationSeverityType
- ReprintGroundCloseDocumentsOptionType
- ShippingDocumentDispositionType
- ShippingDocumentEMailGroupingType
- ShippingDocumentGroupingType
- ShippingDocumentImageType
- ShippingDocumentNamingType
- ShippingDocumentStockType

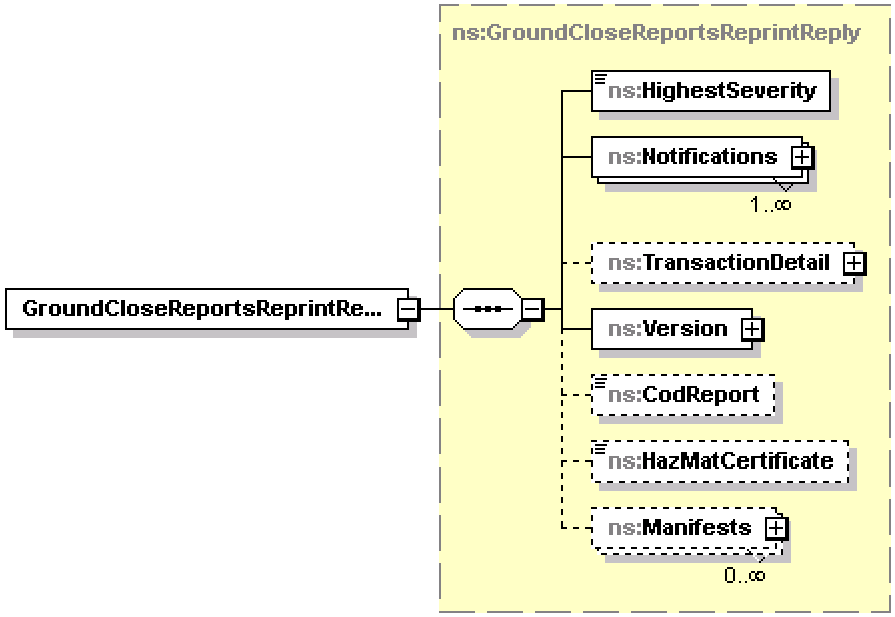
element **GroundCloseDocumentsReply**

diagram	 ns:GroundCloseDocumentsReply GroundCloseDocumentsReply ns:HighestSeverity This indicates the highest level of severity of all the notifications returned in this reply. ns:Notifications 1..∞ The descriptive data regarding the results of the submitted transaction. ns:TransactionDetail Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply. ns:Version Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply). ns:CloseDocuments 0..∞ The actual document contents for all provided reports.
namespace	http://fedex.com/ws/close/v2
type	ns:GroundCloseDocumentsReply
properties	content complex
children	**ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version ns:CloseDocuments**
source	<xs:element name="GroundCloseDocumentsReply" type="ns:GroundCloseDocumentsReply"/>

element **GroundCloseReply**

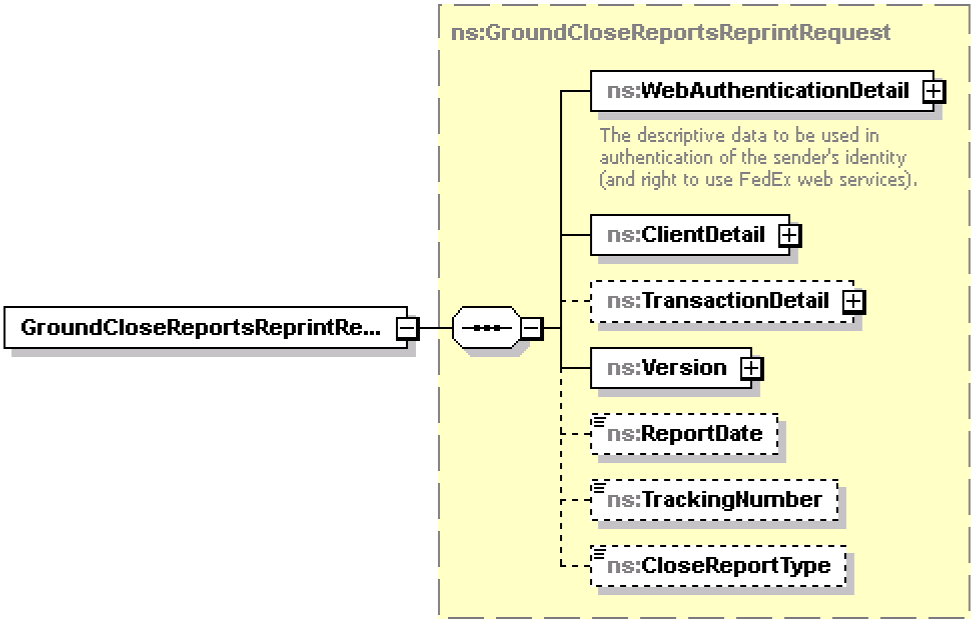
diagram	 ns:GroundCloseReply GroundCloseReply ns:HighestSeverity ns:Notifications 1..∞ ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate ns:Manifest ns:MultiweightReport
namespace	http://fedex.com/ws/close/v2
type	ns:GroundCloseReply
properties	content complex
children	**ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate ns:Manifest ns:MultiweightReport**
source	<xs:element name="GroundCloseReply" type="ns:GroundCloseReply"/>

element **GroundCloseReportsReprintReply**

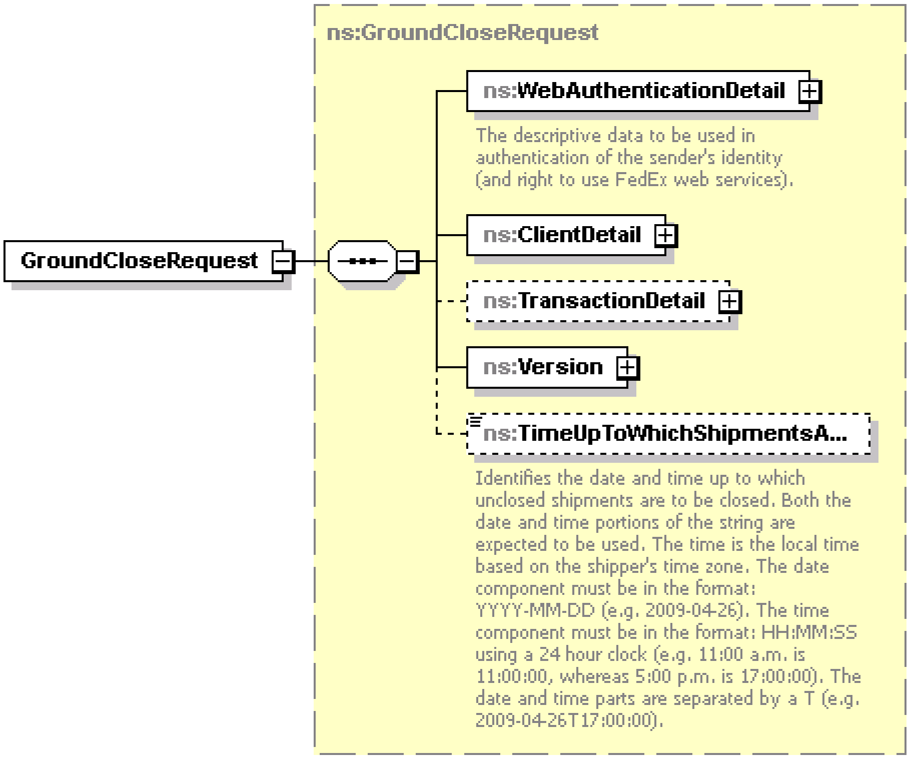
diagram	 ns:GroundCloseReportsReprintReply GroundCloseReportsReprintRe... ns:HighestSeverity ns:Notifications 1..∞ ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate ns:Manifests 0..∞
namespace	http://fedex.com/ws/close/v2
type	ns:GroundCloseReportsReprintReply
properties	content complex
children	**ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate**

	ns:Manifests
source	<xs:element name="GroundCloseReportsReprintReply" type="ns:GroundCloseReportsReprintReply"/>

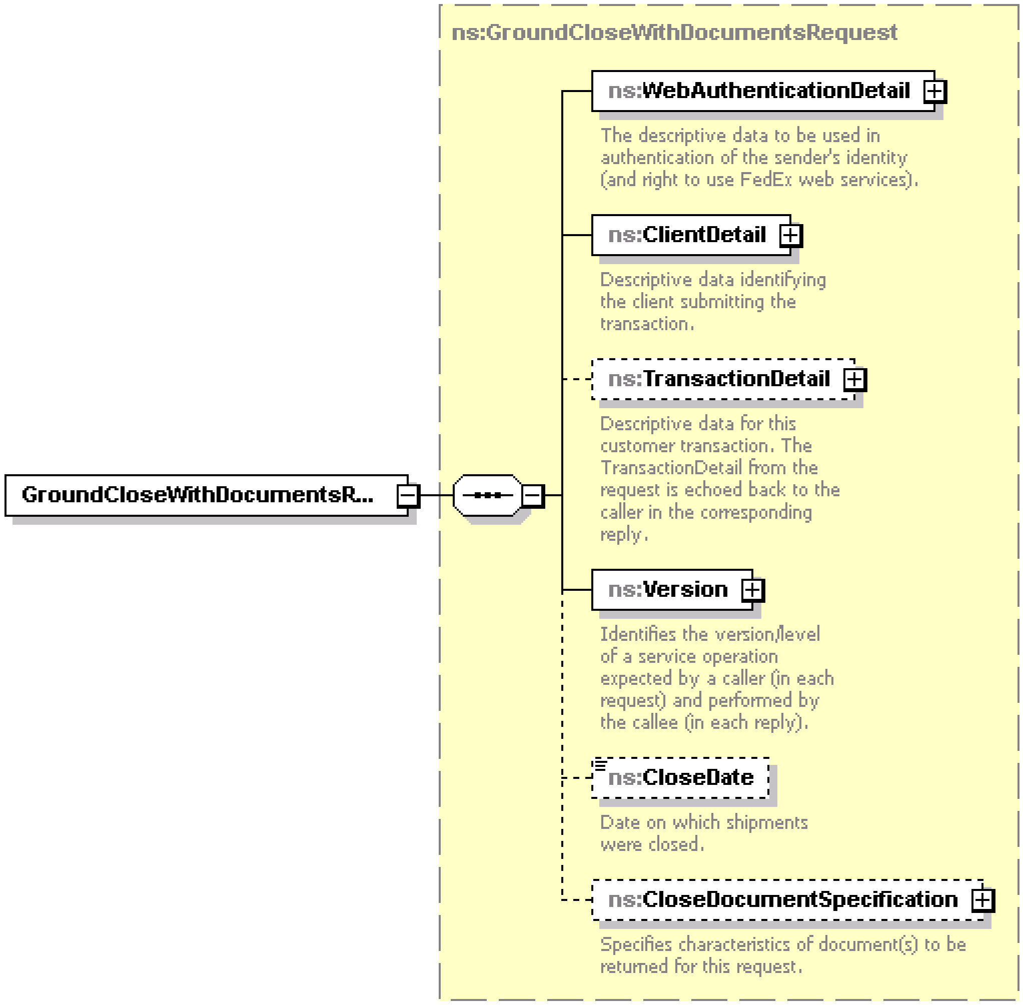
element **GroundCloseReportsReprintRequest**

diagram	 ns:GroundCloseReportsReprintRequest GroundCloseReportsReprintRe... ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:ClientDetail ns:TransactionDetail ns:Version ns:ReportDate ns:TrackingNumber ns:CloseReportType
namespace	http://fedex.com/ws/close/v2
type	ns:GroundCloseReportsReprintRequest
properties	content complex
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:ReportDate ns:TrackingNumber ns:CloseReportType**
source	<xs:element name="GroundCloseReportsReprintRequest" type="ns:GroundCloseReportsReprintRequest"/>

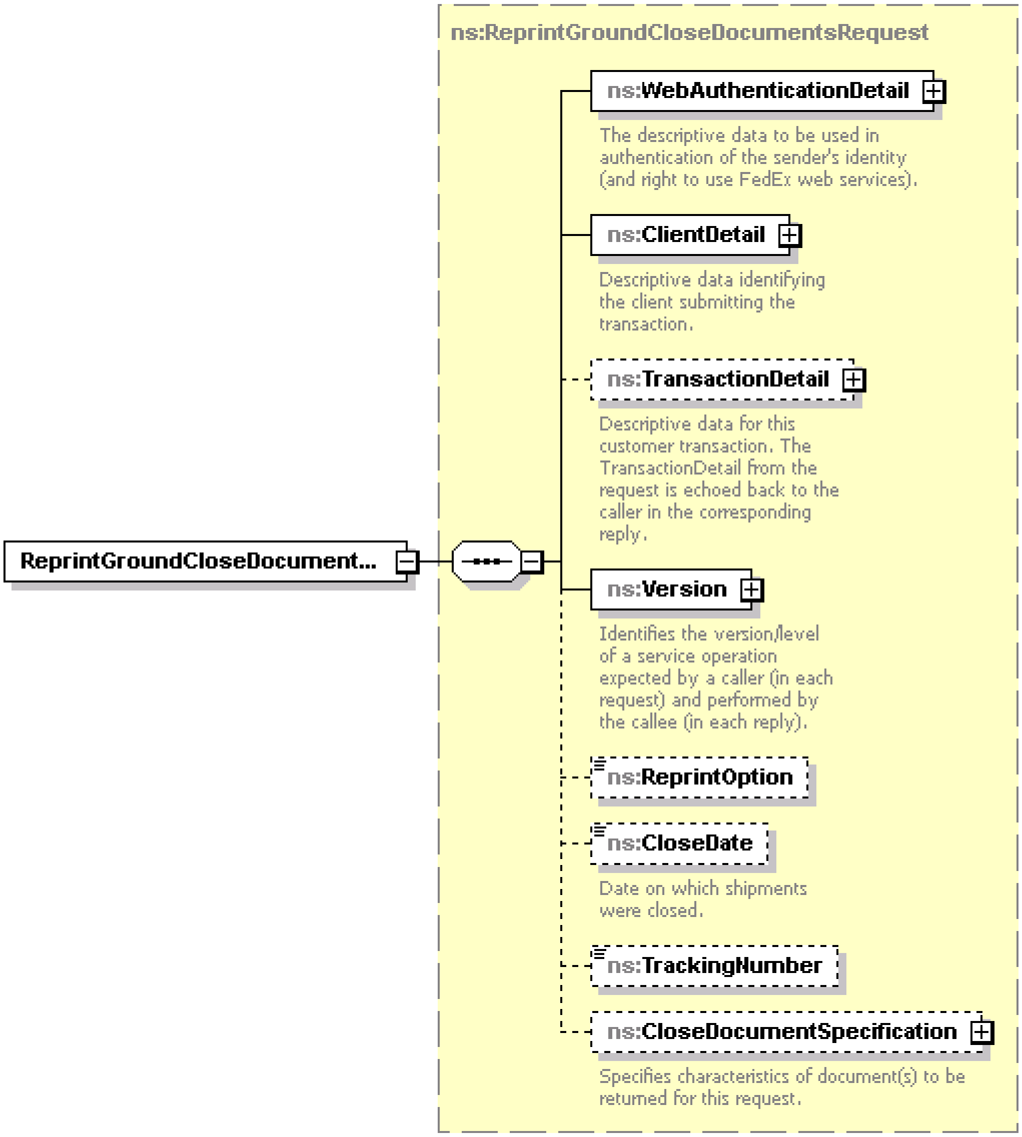
element **GroundCloseRequest**

diagram	 ns:GroundCloseRequest GroundCloseRequest ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:ClientDetail ns:TransactionDetail ns:Version ns:TimeUpToWhichShipmentsA... Identifies the date and time up to which unclosed shipments are to be closed. Both the date and time portions of the string are expected to be used. The time is the local time based on the shipper's time zone. The date component must be in the format: YYYY-MM-DD (e.g. 2009-04-26). The time component must be in the format: HH:MM:SS using a 24 hour clock (e.g. 11:00 a.m. is 11:00:00, whereas 5:00 p.m. is 17:00:00). The date and time parts are separated by a T (e.g. 2009-04-26T17:00:00).
namespace	http://fedex.com/ws/close/v2
type	**ns:GroundCloseRequest**
properties	content complex
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:TimeUpToWhichShipmentsAreToBeClosed**
source	<xs:element name="GroundCloseRequest" type="ns:GroundCloseRequest"/>

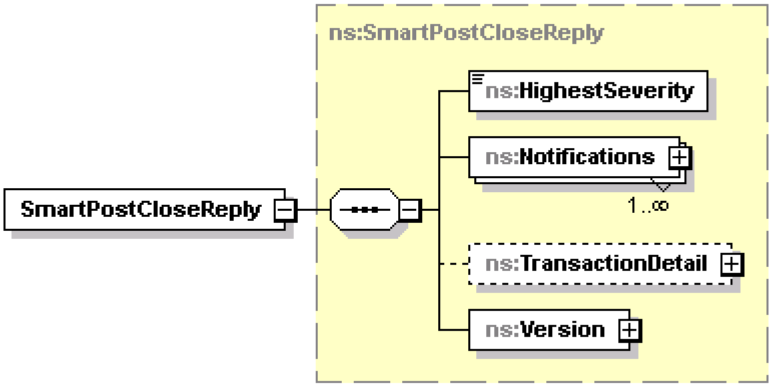
element **GroundCloseWithDocumentsRequest**

diagram	 ns:GroundCloseWithDocumentsRequest GroundCloseWithDocumentsR... ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:ClientDetail Descriptive data identifying the client submitting the transaction. ns:TransactionDetail Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply. ns:Version Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply). ns:CloseDate Date on which shipments were closed. ns:CloseDocumentSpecification Specifies characteristics of document(s) to be returned for this request.
namespace	http://fedex.com/ws/close/v2
type	**ns:GroundCloseWithDocumentsRequest**
properties	content complex
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:CloseDate ns:CloseDocumentSpecification**
source	<xs:element name="GroundCloseWithDocumentsRequest" type="ns:GroundCloseWithDocumentsRequest"/>

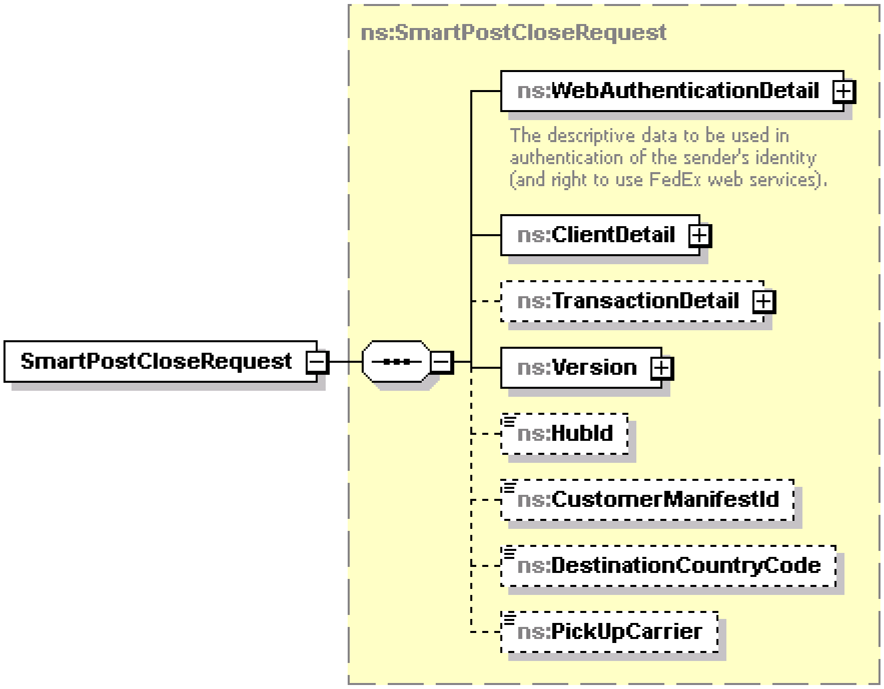
element **ReprintGroundCloseDocumentsRequest**

diagram	 ns:ReprintGroundCloseDocumentsRequest ReprintGroundCloseDocument... ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:ClientDetail Descriptive data identifying the client submitting the transaction. ns:TransactionDetail Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply. ns:Version Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply). ns:ReprintOption ns:CloseDate Date on which shipments were closed. ns:TrackingNumber ns:CloseDocumentSpecification Specifies characteristics of document(s) to be returned for this request.
namespace	http://fedex.com/ws/close/v2
type	**ns:ReprintGroundCloseDocumentsRequest**
properties	content complex
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:ReprintOption ns:CloseDate ns:TrackingNumber ns:CloseDocumentSpecification**
source	<xs:element name="ReprintGroundCloseDocumentsRequest" type="ns:ReprintGroundCloseDocumentsRequest"/>

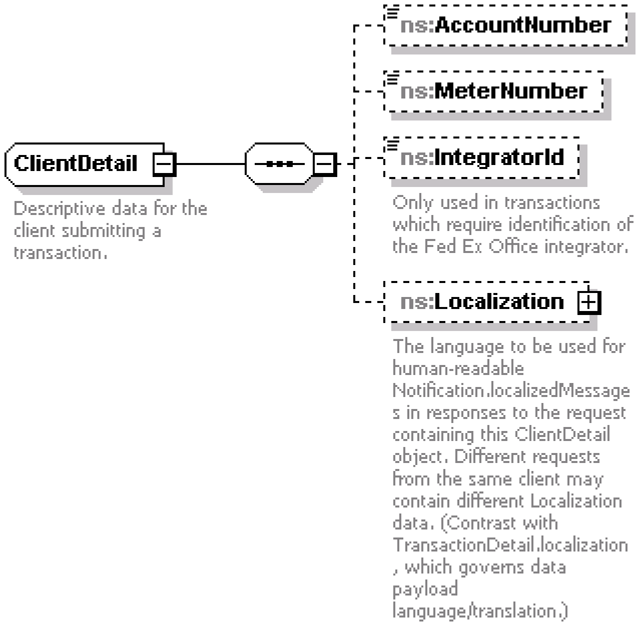
element **SmartPostCloseReply**

diagram	 ns:SmartPostCloseReply SmartPostCloseReply ns:HighestSeverity ns:Notifications 1..∞ ns:TransactionDetail ns:Version
namespace	http://fedex.com/ws/close/v2
type	ns:SmartPostCloseReply
properties	content complex
children	**ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version**
source	<xs:element name="SmartPostCloseReply" type="ns:SmartPostCloseReply"/>

element **SmartPostCloseRequest**

diagram	 ns:SmartPostCloseRequest SmartPostCloseRequest ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:ClientDetail ns:TransactionDetail ns:Version ns:HubId ns:CustomerManifestId ns:DestinationCountryCode ns:PickUpCarrier
namespace	http://fedex.com/ws/close/v2
type	ns:SmartPostCloseRequest
properties	content complex
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:HubId ns:CustomerManifestId ns:DestinationCountryCode ns:PickUpCarrier**
source	<xs:element name="SmartPostCloseRequest" type="ns:SmartPostCloseRequest"/>

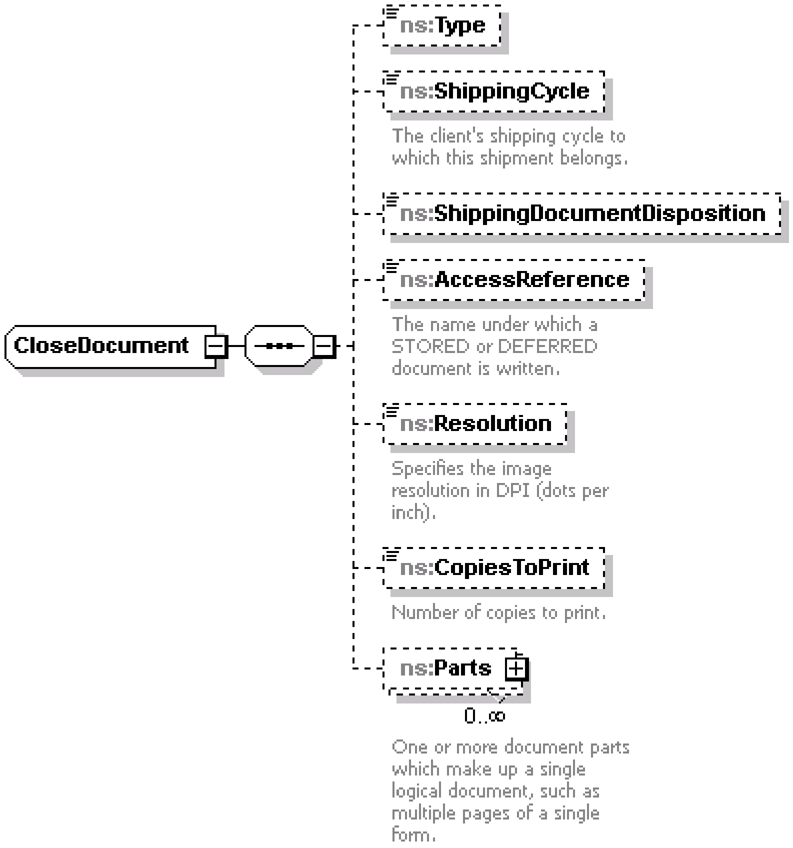
complexType **ClientDetail**

diagram	 ClientDetail Descriptive data for the client submitting a transaction. ns:AccountNumber ns:MeterNumber ns:IntegratorId Only used in transactions which require identification of the Fed Ex Office integrator. ns:Localization 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.)
namespace	http://fedex.com/ws/close/v2
children	**ns:AccountNumber ns:MeterNumber ns:IntegratorId ns:Localization**
used by	elements **GroundCloseReportsReprintRequest/ClientDetail GroundCloseRequest/ClientDetail GroundCloseWithDocumentsRequest/ClientDetail ReprintGroundCloseDocumentsRequest/ClientDetail SmartPostCloseRequest/ClientDetail**
annotation	documentation Descriptive data for the client submitting a transaction.
source	

```
<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="0"/>
    <xs:element name="MeterNumber" type="xs:string" minOccurs="0"/>
    <xs:element name="IntegratorId" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Only used in transactions which require identification of the Fed Ex 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>
  </xs:sequence>
</xs:complexType>
```

complexType **CloseDocument**

diagram


CloseDocument
ns:Type
ns:ShippingCycle
The client's shipping cycle to which this shipment belongs.
ns:ShippingDocumentDisposition
ns:AccessReference
The name under which a STORED or DEFERRED document is written.
ns:Resolution
Specifies the image resolution in DPI (dots per inch).
ns:CopiesToPrint
Number of copies to print.
ns:Parts
0..∞
One or more document parts which make up a single logical document, such as multiple pages of a single form.

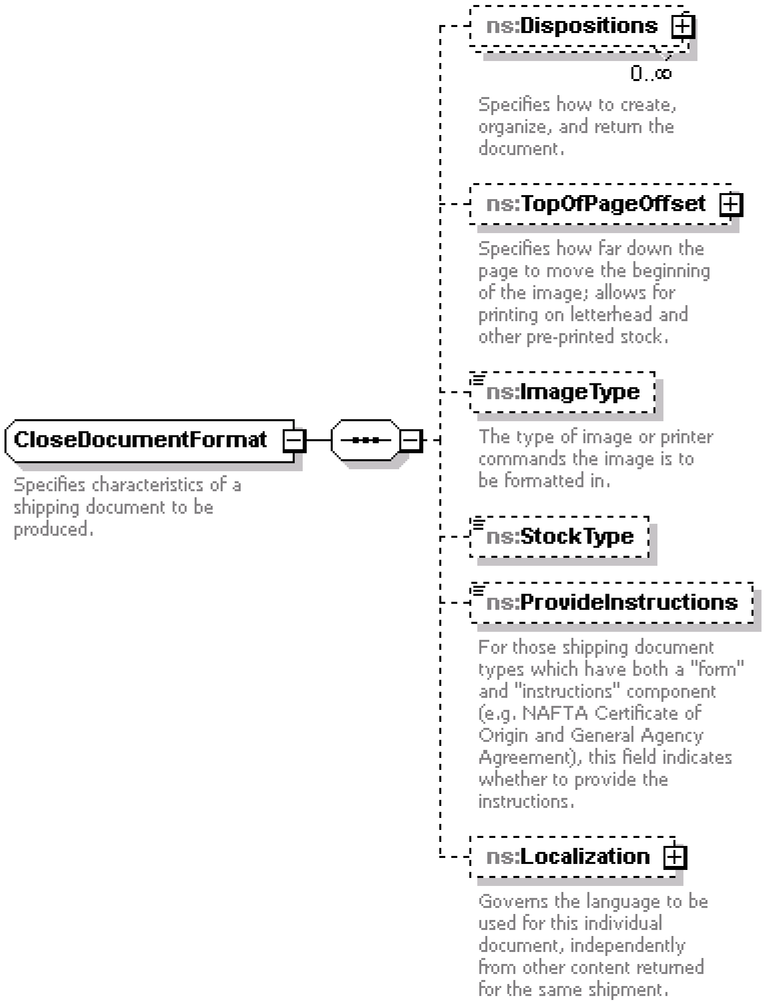
namespace	http://fedex.com/ws/close/v2
children	**ns:Type ns:ShippingCycle ns:ShippingDocumentDisposition ns:AccessReference ns:Resolution ns:CopiesToPrint ns:Parts**
used by	element **GroundCloseDocumentsReply/CloseDocuments**

source

```
<xs:complexType name="CloseDocument">
  <xs:sequence>
    <xs:element name="Type" type="ns:CloseDocumentType" minOccurs="0"/>
    <xs:element name="ShippingCycle" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The client's shipping cycle to which this shipment
belongs.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ShippingDocumentDisposition"
type="ns:ShippingDocumentDispositionType" minOccurs="0"/>
    <xs:element name="AccessReference" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The name under which a STORED or DEFERRED document is
written.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Resolution" type="xs:nonNegativeInteger" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Specifies the image resolution in DPI (dots per inch).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="CopiesToPrint" type="xs:positiveInteger" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Number of copies to print.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Parts" type="ns:ShippingDocumentPart" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>One or more document parts which make up a single logical document,
such as multiple pages of a single form.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **CloseDocumentFormat**

diagram


ns:Dispositions
0..∞
Specifies how to create, organize, and return the document.
ns:TopOfPageOffset
Specifies how far down the page to move the beginning of the image; allows for printing on letterhead and other pre-printed stock.
ns:ImageType
The type of image or printer commands the image is to be formatted in.
ns:StockType
ns:ProvideInstructions
For those shipping document types which have both a "form" and "instructions" component (e.g. NAFTA Certificate of Origin and General Agency Agreement), this field indicates whether to provide the instructions.
ns:Localization
Governs the language to be used for this individual document, independently from other content returned for the same shipment.
CloseDocumentFormat
Specifies characteristics of a shipping document to be produced.

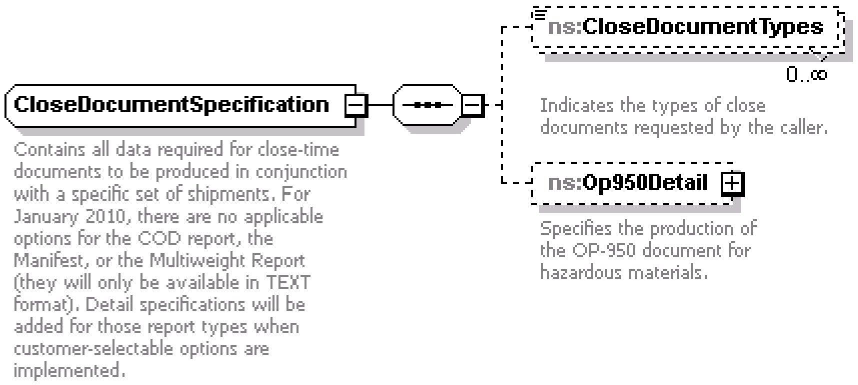
namespace	http://fedex.com/ws/close/v2
children	**ns:Dispositions ns:TopOfPageOffset ns:ImageType ns:StockType ns:ProvideInstructions ns:Localization**
used by	element **Op950Detail/Format**
annotation	documentation Specifies characteristics of a shipping document to be produced.

source

```
<xs:complexType name="CloseDocumentFormat">
  <xs:annotation>
    <xs:documentation>Specifies characteristics of a shipping document to be
produced.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Dispositions" type="ns:ShippingDocumentDispositionDetail" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
```

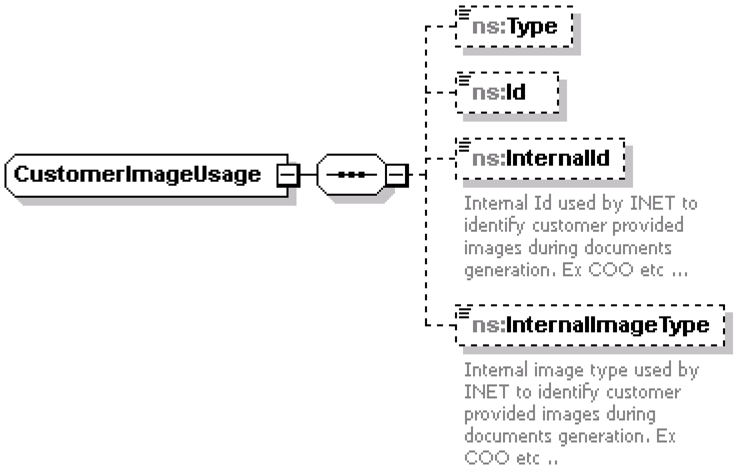
```
        <xs:documentation>Specifies how to create, organize, and return the
document.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TopOfPageOffset" type="ns:LinearMeasure" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Specifies how far down the page to move the beginning of the image;
allows for printing on letterhead and other pre-printed stock.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ImageType" type="ns:ShippingDocumentImageType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The type of image or printer commands the image is to be formatted
in.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="StockType" type="ns:ShippingDocumentStockType" minOccurs="0"/>
    <xs:element name="ProvideInstructions" type="xs:boolean" minOccurs="0">
      <xs:annotation>
        <xs:documentation>For those shipping document types which have both a "form" and
"instructions" component (e.g. NAFTA Certificate of Origin and General Agency Agreement), this
field indicates whether to provide the instructions.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Localization" type="ns:Localization" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Governs the language to be used for this individual document,
independently from other content returned for the same shipment.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **CloseDocumentSpecification**

diagram	 ns:CloseDocumentTypes 0..∞ Indicates the types of close documents requested by the caller. CloseDocumentSpecification Contains all data required for close-time documents to be produced in conjunction with a specific set of shipments. For January 2010, there are no applicable options for the COD report, the Manifest, or the Multiweight Report (they will only be available in TEXT format). Detail specifications will be added for those report types when customer-selectable options are implemented. ns:Op950Detail Specifies the production of the OP-950 document for hazardous materials.
namespace	http://fedex.com/ws/close/v2
children	**ns:CloseDocumentTypes ns:Op950Detail**
used by	elements **GroundCloseWithDocumentsRequest/CloseDocumentSpecification** **ReprintGroundCloseDocumentsRequest/CloseDocumentSpecification**
annotation	documentation Contains all data required for close-time documents to be produced in conjunction with a specific set of shipments. For January 2010, there are no applicable options for the COD report, the Manifest, or the Multiweight Report (they will only be available in TEXT format). Detail specifications will be added for those report types when customer-selectable options are implemented.
source	<xs:complexType name="CloseDocumentSpecification"> <xs:annotation> <xs:documentation>Contains all data required for close-time documents to be produced in conjunction with a specific set of shipments. For January 2010, there are no applicable options for

```
the COD report, the Manifest, or the Multiweight Report (they will only be available in TEXT format).
Detail specifications will be added for those report types when customer-selectable options are
implemented.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="CloseDocumentTypes" type="ns:CloseDocumentType" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Indicates the types of close documents requested by the
caller.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Op950Detail" type="ns:Op950Detail" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Specifies the production of the OP-950 document for hazardous
materials.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **CustomerImageUsage**

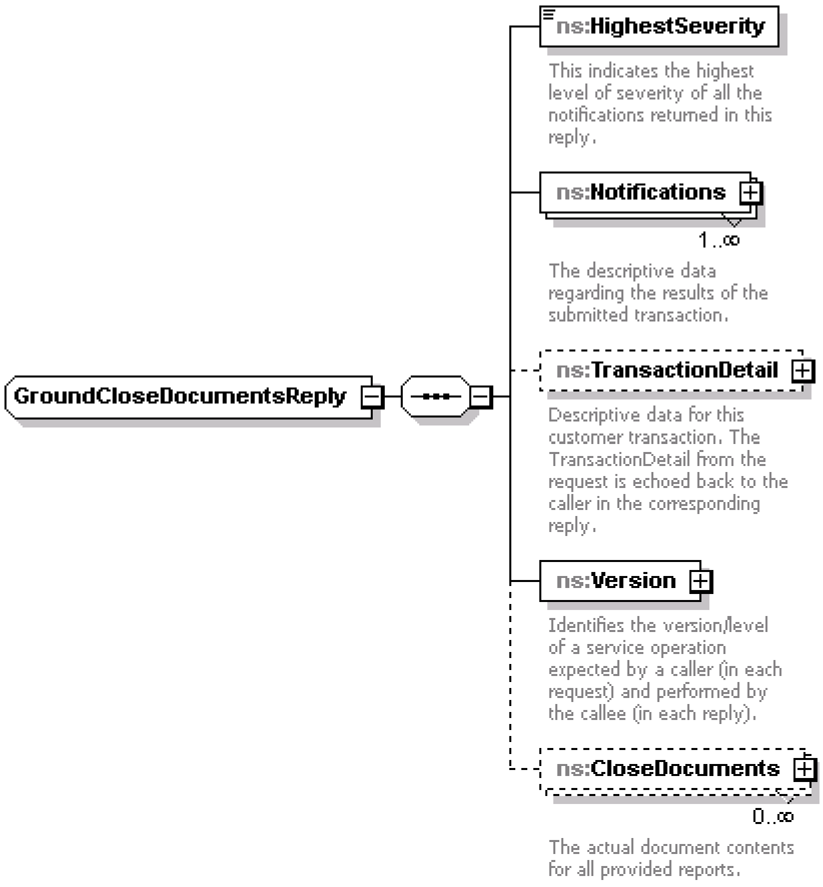
diagram	 ns:Type ns:Id CustomerImageUsage ns:InternalId Internal Id used by INET to identify customer provided images during documents generation. Ex COO etc ... ns:InternalImageType Internal image type used by INET to identify customer provided images during documents generation. Ex COO etc ..
namespace	http://fedex.com/ws/close/v2
children	**ns:Type ns:Id ns:InternalId ns:InternalImageType**
used by	element **Op950Detail/CustomerImageUsages**

source

```
<xs:complexType name="CustomerImageUsage">
  <xs:sequence>
    <xs:element name="Type" type="ns:CustomerImageUsageType" minOccurs="0"/>
    <xs:element name="Id" type="ns:ImageId" minOccurs="0"/>
    <xs:element name="InternalId" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Internal Id used by INET to identify customer provided images during
documents generation. Ex COO etc ...</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="InternalImageType" type="ns:InternalImageType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Internal image type used by INET to identify customer provided images
during documents generation. Ex COO etc ..</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
```

```
</xs:complexType>
```

complexType **GroundCloseDocumentsReply**

diagram	
namespace	http://fedex.com/ws/close/v2
children	**ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version ns:CloseDocuments**
used by	element **GroundCloseDocumentsReply**

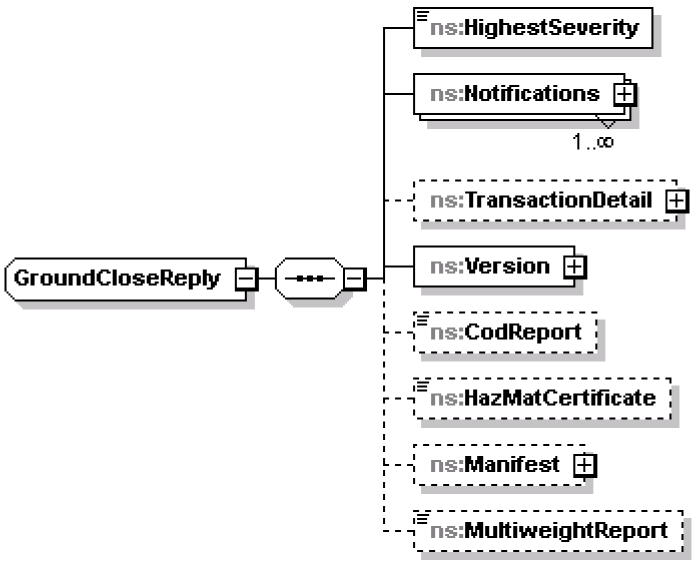
GroundCloseDocumentsReply
ns:HighestSeverity
This indicates the highest level of severity of all the notifications returned in this reply.
ns:Notifications
1..∞
The descriptive data regarding the results of the submitted transaction.
ns:TransactionDetail
Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.
ns:Version
Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).
ns:CloseDocuments
0..∞
The actual document contents for all provided reports.

source

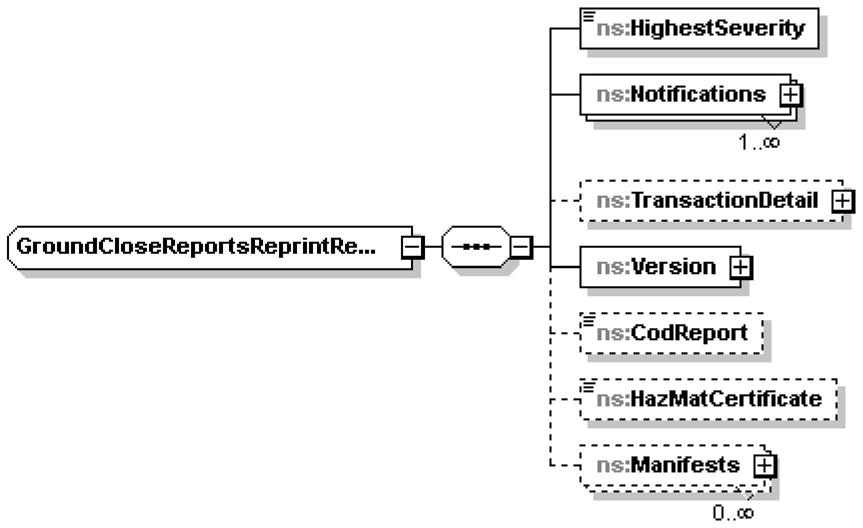
```
<xs:complexType name="GroundCloseDocumentsReply">
  <xs:sequence>
    <xs:element name="HighestSeverity" type="ns:NotificationSeverityType">
      <xs:annotation>
        <xs:documentation>This indicates the highest level of severity of all the notifications returned
in this reply.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Notifications" type="ns:Notification" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>The descriptive data regarding the results of the submitted
transaction.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Descriptive data for this customer transaction. The TransactionDetail from
the request is echoed back to the caller in the corresponding reply.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Version" type="ns: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:element> <xs:element name="CloseDocuments" type="ns:CloseDocument" minOccurs="0" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>The actual document contents for all provided reports.</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType>

complexType **GroundCloseReply**

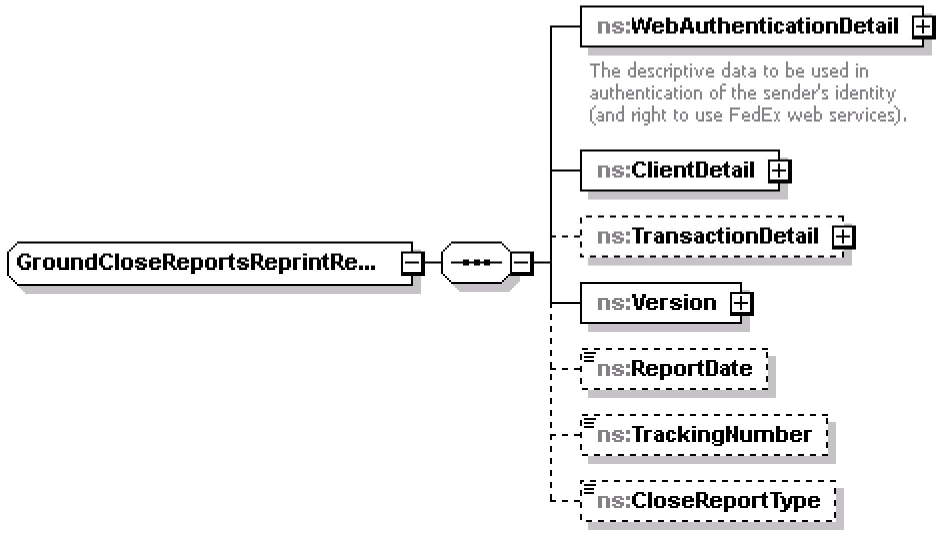
diagram	 GroundCloseReply ns:HighestSeverity ns:Notifications 1..∞ ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate ns:Manifest ns:MultiweightReport
namespace	http://fedex.com/ws/close/v2
children	**ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate ns:Manifest ns:MultiweightReport**
used by	element **GroundCloseReply**
source	<xs:complexType name="GroundCloseReply"> <xs:sequence> <xs:element name="HighestSeverity" type="ns:NotificationSeverityType"/> <xs:element name="Notifications" type="ns:Notification" maxOccurs="unbounded"/> <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"/> <xs:element name="Version" type="ns:VersionId"/> <xs:element name="CodReport" type="xs:base64Binary" minOccurs="0"/> <xs:element name="HazMatCertificate" type="xs:base64Binary" minOccurs="0"/> <xs:element name="Manifest" type="ns:ManifestFile" minOccurs="0"/> <xs:element name="MultiweightReport" type="xs:base64Binary" minOccurs="0"/> </xs:sequence> </xs:complexType>

complexType **GroundCloseReportsReprintReply**

diagram	 GroundCloseReportsReprintRe... ns:HighestSeverity ns:Notifications 1..∞ ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate ns:Manifests 0..∞
namespace	http://fedex.com/ws/close/v2
children	**ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version ns:CodReport ns:HazMatCertificate ns:Manifests**
used by	element **GroundCloseReportsReprintReply**

source	<xs:complexType name="GroundCloseReportsReprintReply">
	  <xs:sequence>
	    <xs:element name="HighestSeverity" type="ns:NotificationSeverityType"/>
	    <xs:element name="Notifications" type="ns:Notification" maxOccurs="unbounded"/>
	    <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"/>
	    <xs:element name="Version" type="ns:VersionId"/>
	    <xs:element name="CodReport" type="xs:base64Binary" minOccurs="0"/>
	    <xs:element name="HazMatCertificate" type="xs:base64Binary" minOccurs="0"/>
	    <xs:element name="Manifests" type="ns:ManifestFile" minOccurs="0" maxOccurs="unbounded"/>
	  </xs:sequence>
	</xs:complexType>

complexType **GroundCloseReportsReprintRequest**

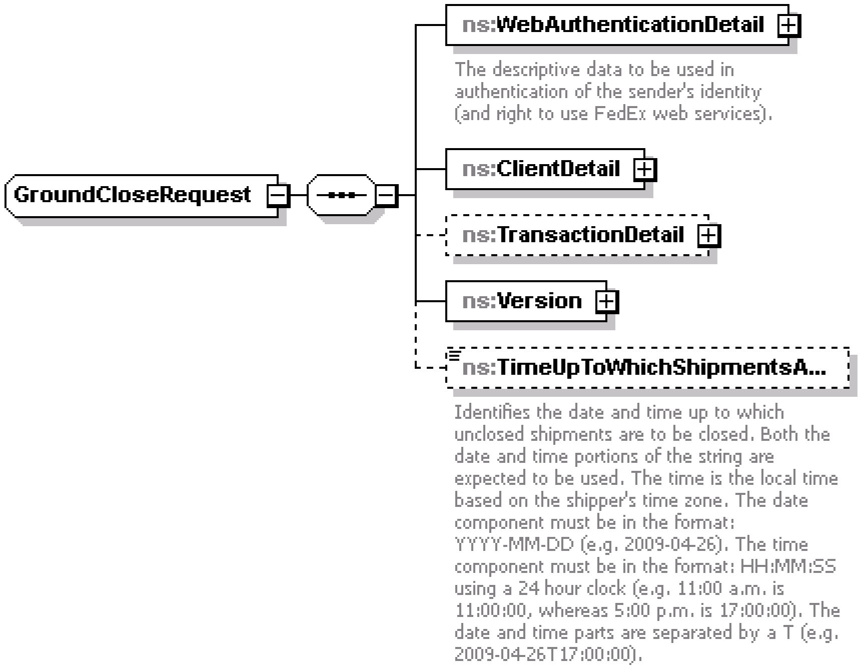
diagram	 GroundCloseReportsReprintRe... ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:ClientDetail ns:TransactionDetail ns:Version ns:ReportDate ns:TrackingNumber ns:CloseReportType
namespace	http://fedex.com/ws/close/v2
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:ReportDate ns:TrackingNumber**

	ns:CloseReportType
used by	element **GroundCloseReportsReprintRequest**

source

```
<xs:complexType name="GroundCloseReportsReprintRequest">
  <xs:sequence>
    <xs:element name="WebAuthenticationDetail" type="ns:WebAuthenticationDetail">
      <xs:annotation>
        <xs:documentation>The 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:element name="ClientDetail" type="ns:ClientDetail"/>
    <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"/>
    <xs:element name="Version" type="ns:VersionId"/>
    <xs:element name="ReportDate" type="xs:date" minOccurs="0"/>
    <xs:element name="TrackingNumber" type="xs:string" minOccurs="0"/>
    <xs:element name="CloseReportType" type="ns:CloseReportType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

complexType **GroundCloseRequest**

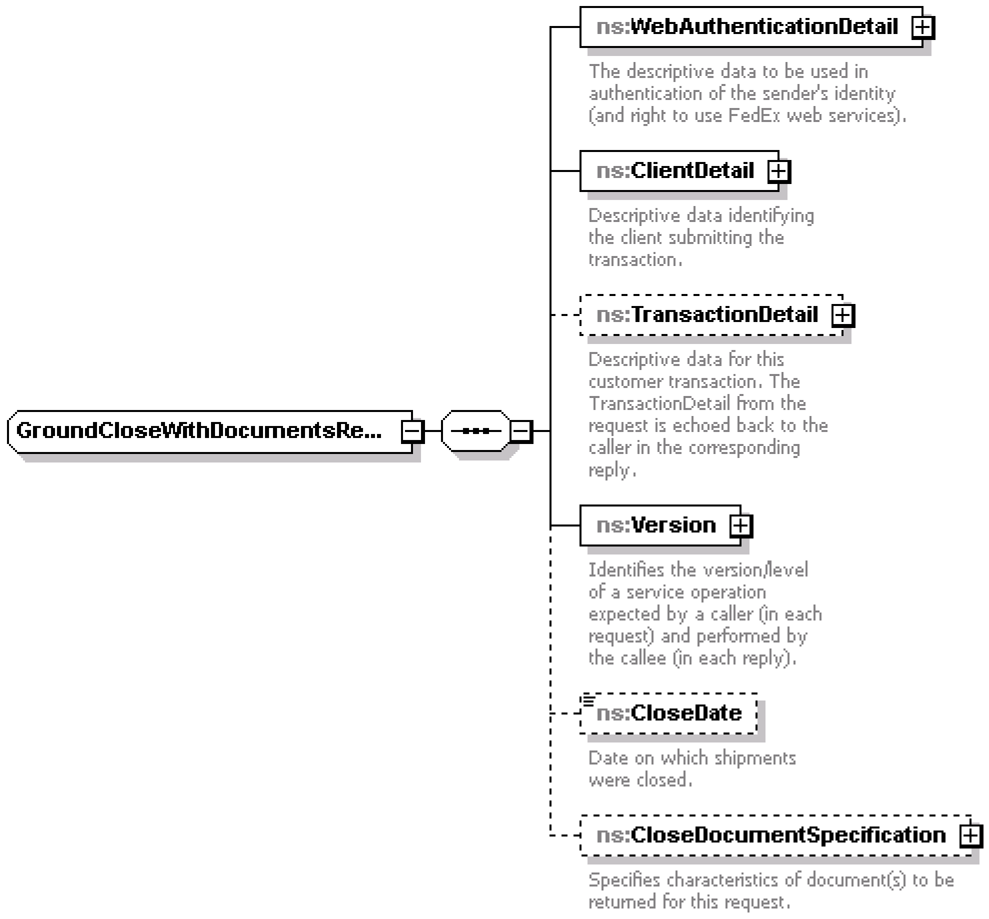
diagram	 GroundCloseRequest ns:WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:ClientDetail ns:TransactionDetail ns:Version ns:TimeUpToWhichShipmentsA... Identifies the date and time up to which unclosed shipments are to be closed. Both the date and time portions of the string are expected to be used. The time is the local time based on the shipper's time zone. The date component must be in the format: YYYY-MM-DD (e.g. 2009-04-26). The time component must be in the format: HH:MM:SS using a 24 hour clock (e.g. 11:00 a.m. is 11:00:00, whereas 5:00 p.m. is 17:00:00). The date and time parts are separated by a T (e.g. 2009-04-26T17:00:00).
namespace	http://fedex.com/ws/close/v2
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:TimeUpToWhichShipmentsAreToBeClosed**
used by	element **GroundCloseRequest**

source

```
<xs:complexType name="GroundCloseRequest">
  <xs:sequence>
    <xs:element name="WebAuthenticationDetail" type="ns:WebAuthenticationDetail">
      <xs:annotation>
        <xs:documentation>The 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:element name="ClientDetail" type="ns:ClientDetail"/>
    <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"/>
    <xs:element name="Version" type="ns:VersionId"/>
    <xs:element name="TimeUpToWhichShipmentsAreToBeClosed" type="xs:dateTime"
minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the date and time up to which unclosed shipments are to be
closed. Both the date and time portions of the string are expected to be used. The time is the local
time based on the shipper's time zone. The date component must be in the format: YYYY-MM-DD
(e.g. 2009-04-26). The time component must be in the format: HH:MM:SS using a 24 hour clock
(e.g. 11:00 a.m. is 11:00:00, whereas 5:00 p.m. is 17:00:00). The date and time parts are
separated by a T (e.g. 2009-04-26T17:00:00).</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **GroundCloseWithDocumentsRequest**

diagram


GroundCloseWithDocumentsRe...
ns:WebAuthenticationDetail
The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services).
ns:ClientDetail
Descriptive data identifying the client submitting the transaction.
ns:TransactionDetail
Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.
ns:Version
Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).
ns:CloseDate
Date on which shipments were closed.
ns:CloseDocumentSpecification
Specifies characteristics of document(s) to be returned for this request.

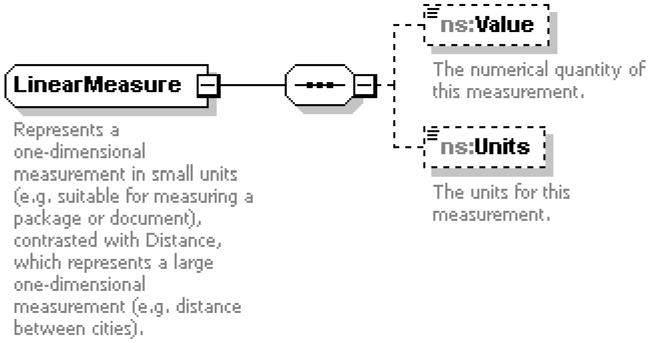
namespace	http://fedex.com/ws/close/v2
children	**ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:CloseDate ns:CloseDocumentSpecification**
used by	element **GroundCloseWithDocumentsRequest**

source

```
<xs:complexType name="GroundCloseWithDocumentsRequest">
  <xs:sequence>
    <xs:element name="WebAuthenticationDetail" type="ns:WebAuthenticationDetail">
      <xs:annotation>
        <xs:documentation>The 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:element name="ClientDetail" type="ns:ClientDetail">
```

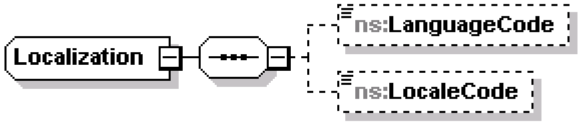
	<xs:annotation> <xs:documentation>Descriptive data identifying the client submitting the transaction.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"> <xs:annotation> <xs:documentation>Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="Version" type="ns: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:element> <xs:element name="CloseDate" type="xs:date" minOccurs="0"> <xs:annotation> <xs:documentation>Date on which shipments were closed.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="CloseDocumentSpecification" type="ns:CloseDocumentSpecification" minOccurs="0"> <xs:annotation> <xs:documentation>Specifies characteristics of document(s) to be returned for this request.</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType>

complexType **LinearMeasure**

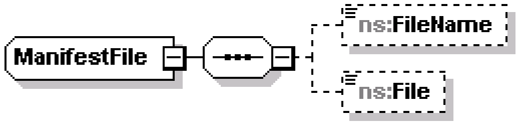
diagram	 ns:Value The numerical quantity of this measurement. LinearMeasure Represents a one-dimensional measurement in small units (e.g. suitable for measuring a package or document), contrasted with Distance, which represents a large one-dimensional measurement (e.g. distance between cities). ns:Units The units for this measurement.
namespace	http://fedex.com/ws/close/v2
children	**ns:Value ns:Units**
used by	element **CloseDocumentFormat/TopOfPageOffset**
annotation	documentation Represents a one-dimensional measurement in small units (e.g. suitable for measuring a package or document), contrasted with Distance, which represents a large one-dimensional measurement (e.g. distance between cities).
source	<xs:complexType name="LinearMeasure"> <xs:annotation> <xs:documentation>Represents a one-dimensional measurement in small units (e.g. suitable for measuring a package or document), contrasted with Distance, which represents a large one-dimensional measurement (e.g. distance between cities).</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="Value" type="xs:decimal" minOccurs="0">

	<xs:annotation> <xs:documentation>The numerical quantity of this measurement.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="Units" type="ns:LinearUnits" minOccurs="0"> <xs:annotation> <xs:documentation>The units for this measurement.</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType>

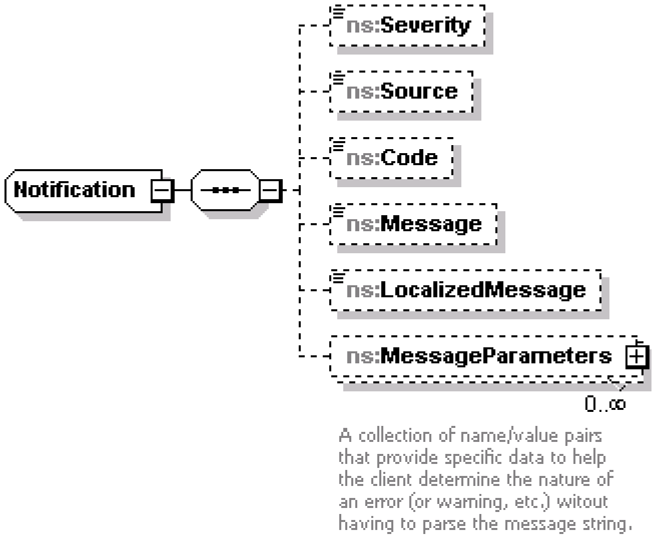
complexType **Localization**

diagram	
namespace	http://fedex.com/ws/close/v2
children	**ns:LanguageCode ns:LocaleCode**
used by	elements **ClientDetail/Localization CloseDocumentFormat/Localization TransactionDetail/Localization**
source	<xs:complexType name="Localization"> <xs:sequence> <xs:element name="LanguageCode" type="xs:string" minOccurs="0"/> <xs:element name="LocaleCode" type="xs:string" minOccurs="0"/> </xs:sequence> </xs:complexType>

complexType **ManifestFile**

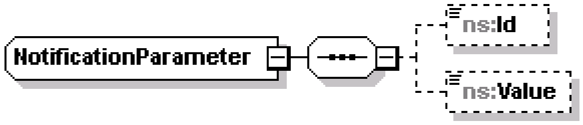
diagram	
namespace	http://fedex.com/ws/close/v2
children	**ns:FileName ns:File**
used by	elements **GroundCloseReply/Manifest GroundCloseReportsReprintReply/Manifests**
source	<xs:complexType name="ManifestFile"> <xs:sequence> <xs:element name="FileName" type="xs:string" minOccurs="0"/> <xs:element name="File" type="xs:base64Binary" minOccurs="0"/> </xs:sequence> </xs:complexType>

complexType **Notification**

diagram	 Notification ns:Severity ns:Source ns:Code ns:Message ns:LocalizedMessage ns:MessageParameters 0..∞ A collection of name/value pairs that provide specific data to help the client determine the nature of an error (or warning, etc.) witout having to parse the message string.
namespace	http://fedex.com/ws/close/v2
children	**ns:Severity ns:Source ns:Code ns:Message ns:LocalizedMessage ns:MessageParameters**
used by	elements **GroundCloseDocumentsReply/Notifications GroundCloseReply/Notifications GroundCloseReportsReprintReply/Notifications SmartPostCloseReply/Notifications**
source	

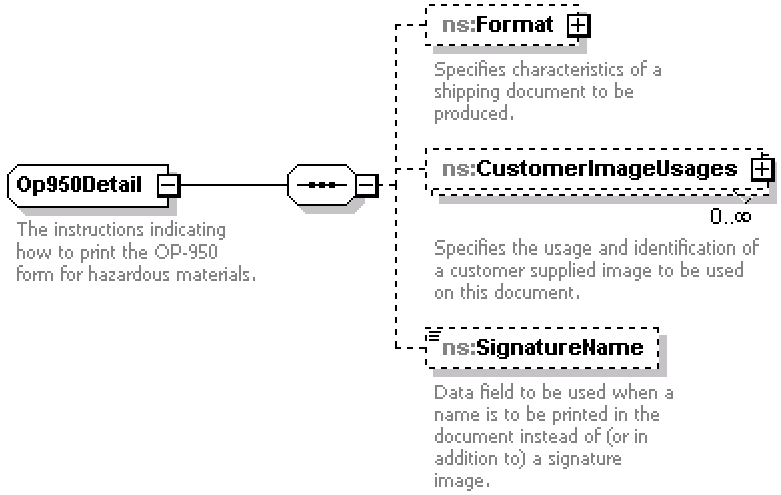
```
<xs:complexType name="Notification">
  <xs:sequence>
    <xs:element name="Severity" type="ns:NotificationSeverityType" minOccurs="0"/>
    <xs:element name="Source" type="xs:string" minOccurs="0"/>
    <xs:element name="Code" type="xs:string" minOccurs="0"/>
    <xs:element name="Message" type="xs:string" minOccurs="0"/>
    <xs:element name="LocalizedMessage" type="xs:string" minOccurs="0"/>
    <xs:element name="MessageParameters" type="ns:NotificationParameter" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>A collection of name/value pairs that provide specific data to help the client determine the nature of an error (or warning, etc.) witout having to parse the message string.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **NotificationParameter**

diagram	
namespace	http://fedex.com/ws/close/v2
children	**ns:Id ns:Value**
used by	element **Notification/MessageParameters**
source	

```
<xs:complexType name="NotificationParameter">
  <xs:sequence>
    <xs:element name="Id" type="xs:string" minOccurs="0"/>
    <xs:element name="Value" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

complexType **Op950Detail**

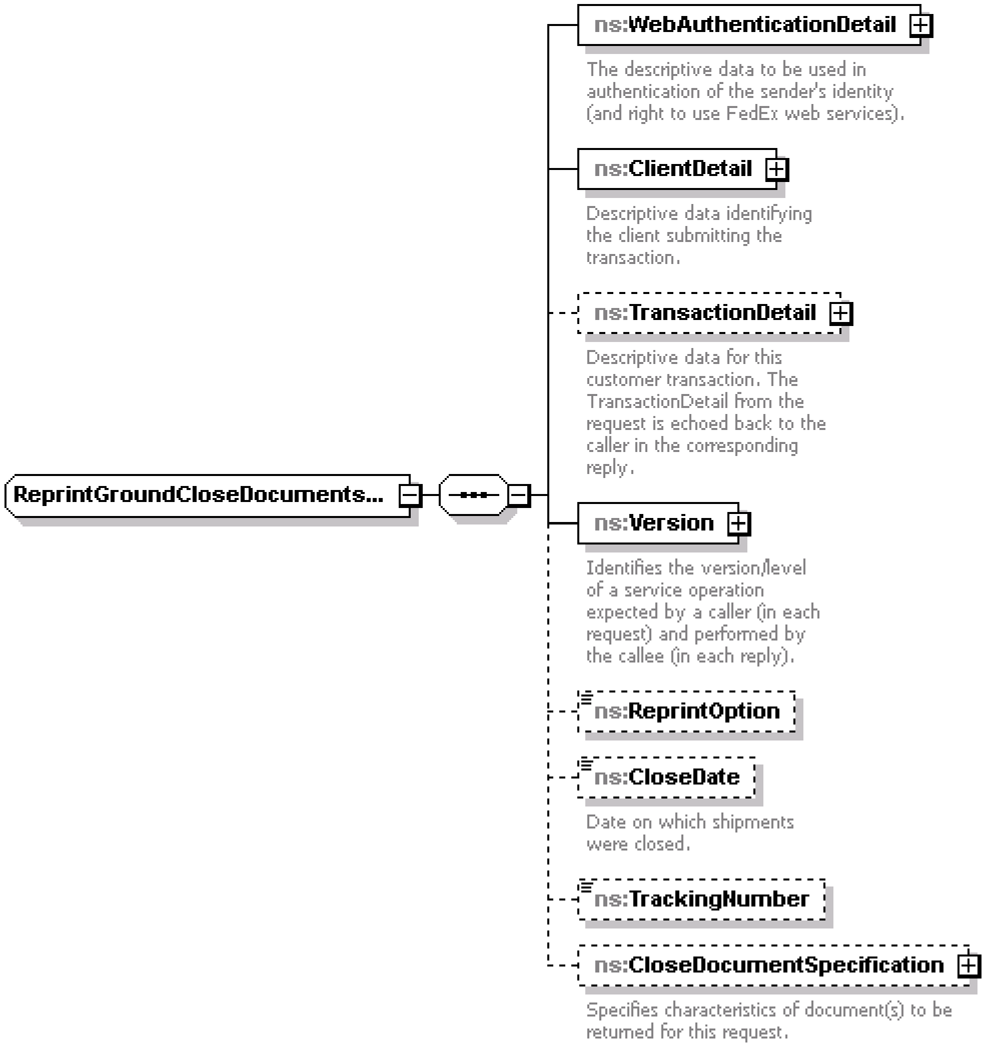
diagram	 ns:Format Specifies characteristics of a shipping document to be produced. Op950Detail The instructions indicating how to print the OP-950 form for hazardous materials. ns:CustomerImageUsages 0..∞ Specifies the usage and identification of a customer supplied image to be used on this document. ns:SignatureName Data field to be used when a name is to be printed in the document instead of (or in addition to) a signature image.
namespace	http://fedex.com/ws/close/v2
children	**ns:Format ns:CustomerImageUsages ns:SignatureName**
used by	element **CloseDocumentSpecification/Op950Detail**
annotation	documentation The instructions indicating how to print the OP-950 form for hazardous materials.

source

```
<xs:complexType name="Op950Detail">
  <xs:annotation>
    <xs:documentation>The instructions indicating how to print the OP-950 form for hazardous materials.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Format" type="ns:CloseDocumentFormat" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Specifies characteristics of a shipping document to be produced.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="CustomerImageUsages" type="ns:CustomerImageUsage" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Specifies the usage and identification of a customer supplied image to be used on this document.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SignatureName" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Data field to be used when a name is to be printed in the document instead of (or in addition to) a signature image.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **ReprintGroundCloseDocumentsRequest**

diagram


ReprintGroundCloseDocuments...
ns:WebAuthenticationDetail
The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services).
ns:ClientDetail
Descriptive data identifying the client submitting the transaction.
ns:TransactionDetail
Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.
ns:Version
Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply).
ns:ReprintOption
ns:CloseDate
Date on which shipments were closed.
ns:TrackingNumber
ns:CloseDocumentSpecification
Specifies characteristics of document(s) to be returned for this request.

namespace

http://fedex.com/ws/close/v2

children

ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:ReprintOption ns:CloseDate ns:TrackingNumber ns:CloseDocumentSpecification

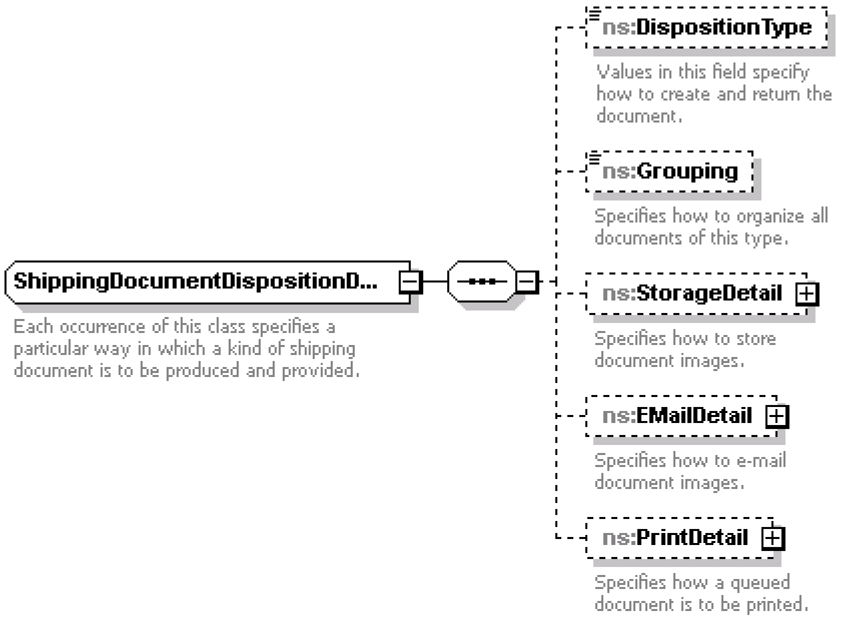
used by

element **ReprintGroundCloseDocumentsRequest**

source

```
<xs:complexType name="ReprintGroundCloseDocumentsRequest">
  <xs:sequence>
    <xs:element name="WebAuthenticationDetail" type="ns:WebAuthenticationDetail">
      <xs:annotation>
        <xs:documentation>The 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:element name="ClientDetail" type="ns:ClientDetail">
      <xs:annotation>
        <xs:documentation>Descriptive data identifying the client submitting the transaction.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Descriptive data for this customer transaction. The TransactionDetail from the request is echoed back to the caller in the corresponding reply.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Version" type="ns: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:element>
    <xs:element name="ReprintOption" type="ns:ReprintGroundCloseDocumentsOptionType"
minOccurs="0"/>
    <xs:element name="CloseDate" type="xs:date" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Date on which shipments were closed.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TrackingNumber" type="xs:string" minOccurs="0"/>
    <xs:element name="CloseDocumentSpecification" type="ns:CloseDocumentSpecification"
minOccurs="0">
      <xs:annotation>
        <xs:documentation>Specifies characteristics of document(s) to be returned for this
request.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **ShippingDocumentDispositionDetail**

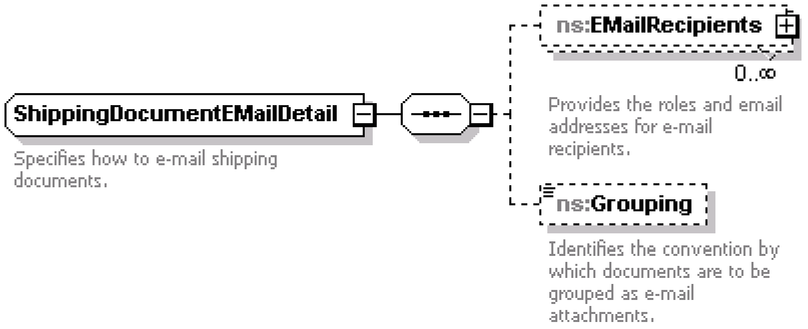
diagram	 ns:DispositionType Values in this field specify how to create and return the document. ns:Grouping Specifies how to organize all documents of this type. ShippingDocumentDispositionD... Each occurrence of this class specifies a particular way in which a kind of shipping document is to be produced and provided. ns:StorageDetail Specifies how to store document images. ns:EMailDetail Specifies how to e-mail document images. ns:PrintDetail Specifies how a queued document is to be printed.
namespace	http://fedex.com/ws/close/v2
children	**ns:DispositionType ns:Grouping ns:StorageDetail ns:EMailDetail ns:PrintDetail**
used by	element **CloseDocumentFormat/Dispositions**
annotation	documentation Each occurrence of this class specifies a particular way in which a kind of shipping document is to be produced and provided.

source

```
<xs:complexType name="ShippingDocumentDispositionDetail">
  <xs:annotation>
    <xs:documentation>Each occurrence of this class specifies a particular way in which a kind of
shipping document is to be produced and provided.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="DispositionType" type="ns:ShippingDocumentDispositionType"
```

```
minOccurs="0">
   <xs:annotation>
    <xs:documentation>Values in this field specify how to create and return the
document.</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="Grouping" type="ns:ShippingDocumentGroupingType" minOccurs="0">
   <xs:annotation>
    <xs:documentation>Specifies how to organize all documents of this type.</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="StorageDetail" type="ns:ShippingDocumentStorageDetail" minOccurs="0">
   <xs:annotation>
    <xs:documentation>Specifies how to store document images.</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="EMailDetail" type="ns:ShippingDocumentEMailDetail" minOccurs="0">
   <xs:annotation>
    <xs:documentation>Specifies how to e-mail document images.</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="PrintDetail" type="ns:ShippingDocumentPrintDetail" minOccurs="0">
   <xs:annotation>
    <xs:documentation>Specifies how a queued document is to be printed.</xs:documentation>
   </xs:annotation>
  </xs:element>
 </xs:sequence>
</xs:complexType>
```

complexType **ShippingDocumentEMailDetail**

diagram	 ns:EMailRecipients 0..∞ Provides the roles and email addresses for e-mail recipients. ShippingDocumentEMailDetail Specifies how to e-mail shipping documents. ns:Grouping Identifies the convention by which documents are to be grouped as e-mail attachments.
namespace	http://fedex.com/ws/close/v2
children	**ns:EMailRecipients ns:Grouping**
used by	element **ShippingDocumentDispositionDetail/EMailDetail**
annotation	documentation Specifies how to e-mail shipping documents.

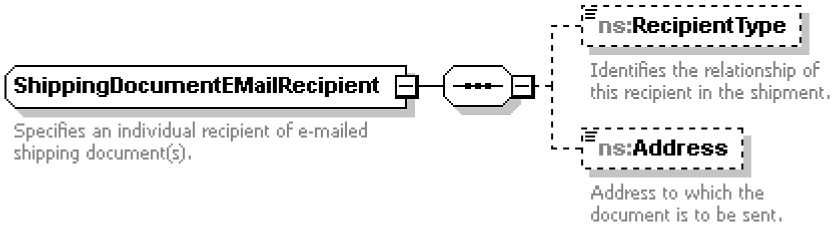
source

```
<xs:complexType name="ShippingDocumentEMailDetail">
 <xs:annotation>
  <xs:documentation>Specifies how to e-mail shipping documents.</xs:documentation>
 </xs:annotation>
 <xs:sequence>
  <xs:element name="EMailRecipients" type="ns:ShippingDocumentEMailRecipient"
minOccurs="0" maxOccurs="unbounded">
   <xs:annotation>
    <xs:documentation>Provides the roles and email addresses for e-mail
recipients.</xs:documentation>
   </xs:annotation>
  </xs:element>
```

```
    <xs:element name="Grouping" type="ns:ShippingDocumentEMailGroupingType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the convention by which documents are to be grouped as e-mail attachments.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **ShippingDocumentEMailRecipient**

diagram


ShippingDocumentEMailRecipient
Specifies an individual recipient of e-mailed shipping document(s).
ns:RecipientType
Identifies the relationship of this recipient in the shipment.
ns:Address
Address to which the document is to be sent.

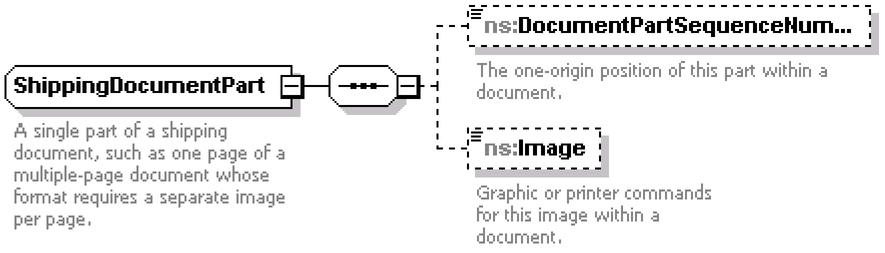
namespace	http://fedex.com/ws/close/v2
children	**ns:RecipientType ns:Address**
used by	element **ShippingDocumentEMailDetail/EMailRecipients**
annotation	documentation Specifies an individual recipient of e-mailed shipping document(s).

source

```
<xs:complexType name="ShippingDocumentEMailRecipient">
  <xs:annotation>
    <xs:documentation>Specifies an individual recipient of e-mailed shipping document(s).</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="RecipientType" type="ns:EMailNotificationRecipientType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the relationship of this recipient in the shipment.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Address" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Address to which the document is to be sent.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **ShippingDocumentPart**

diagram

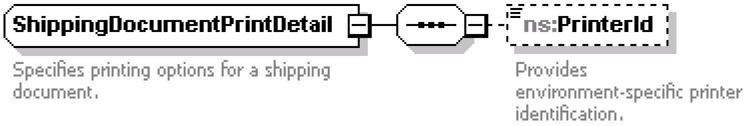

ShippingDocumentPart
A single part of a shipping document, such as one page of a multiple-page document whose format requires a separate image per page.
ns:DocumentPartSequenceNum...
The one-origin position of this part within a document.
ns:Image
Graphic or printer commands for this image within a document.

namespace	http://fedex.com/ws/close/v2

children	**ns:DocumentPartSequenceNumber ns:Image**
used by	element **CloseDocument/Parts**
annotation	documentation A single part of a shipping document, such as one page of a multiple-page document whose format requires a separate image per page.

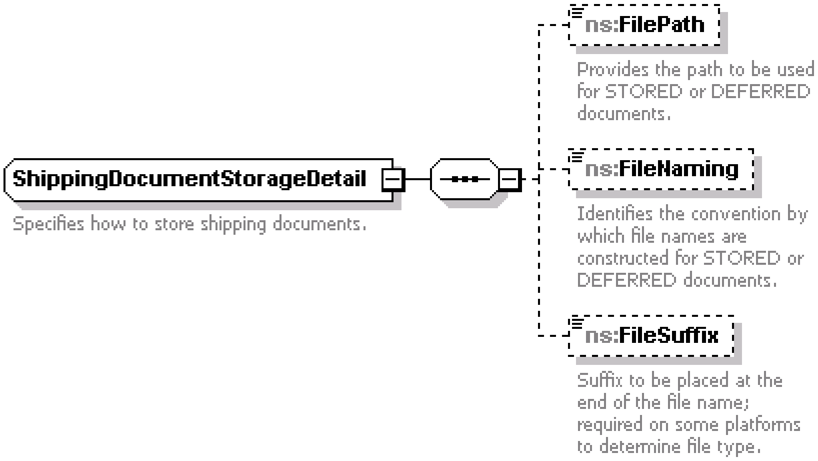
source	`<xs:complexType name="ShippingDocumentPart">`
	`  <xs:annotation>`
	`    <xs:documentation>A single part of a shipping document, such as one page of a multiple-page document whose format requires a separate image per page.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`    <xs:element name="DocumentPartSequenceNumber" type="xs:positiveInteger" minOccurs="0">`
	`      <xs:annotation>`
	`        <xs:documentation>The one-origin position of this part within a document.</xs:documentation>`
	`      </xs:annotation>`
	`    </xs:element>`
	`    <xs:element name="Image" type="xs:base64Binary" minOccurs="0">`
	`      <xs:annotation>`
	`        <xs:documentation>Graphic or printer commands for this image within a document.</xs:documentation>`
	`      </xs:annotation>`
	`    </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

complexType **ShippingDocumentPrintDetail**

diagram	 ShippingDocumentPrintDetail Specifies printing options for a shipping document. ns:PrinterId Provides environment-specific printer identification.
namespace	http://fedex.com/ws/close/v2
children	**ns:PrinterId**
used by	element **ShippingDocumentDispositionDetail/PrintDetail**
annotation	documentation Specifies printing options for a shipping document.

source	`<xs:complexType name="ShippingDocumentPrintDetail">`
	`  <xs:annotation>`
	`    <xs:documentation>Specifies printing options for a shipping document.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`    <xs:element name="PrinterId" type="xs:string" minOccurs="0">`
	`      <xs:annotation>`
	`        <xs:documentation>Provides environment-specific printer identification.</xs:documentation>`
	`      </xs:annotation>`
	`    </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

complexType **ShippingDocumentStorageDetail**

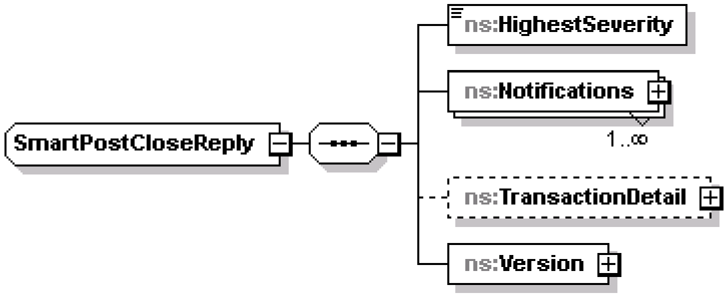
diagram	 ns:FilePath Provides the path to be used for STORED or DEFERRED documents. ShippingDocumentStorageDetail Specifies how to store shipping documents. ns:FileNaming Identifies the convention by which file names are constructed for STORED or DEFERRED documents. ns:FileSuffix Suffix to be placed at the end of the file name; required on some platforms to determine file type.
namespace	http://fedex.com/ws/close/v2
children	**ns:FilePath ns:FileNaming ns:FileSuffix**
used by	element **ShippingDocumentDispositionDetail/StorageDetail**
annotation	documentation Specifies how to store shipping documents.

source

```
<xs:complexType name="ShippingDocumentStorageDetail">
  <xs:annotation>
    <xs:documentation>Specifies how to store shipping documents.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="FilePath" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Provides the path to be used for STORED or DEFERRED documents.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="FileNaming" type="ns:ShippingDocumentNamingType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identifies the convention by which file names are constructed for STORED or DEFERRED documents.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="FileSuffix" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Suffix to be placed at the end of the file name; required on some platforms to determine file type.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

complexType **SmartPostCloseReply**

diagram



namespace http://fedex.com/ws/close/v2

children **ns:HighestSeverity ns:Notifications ns:TransactionDetail ns:Version**

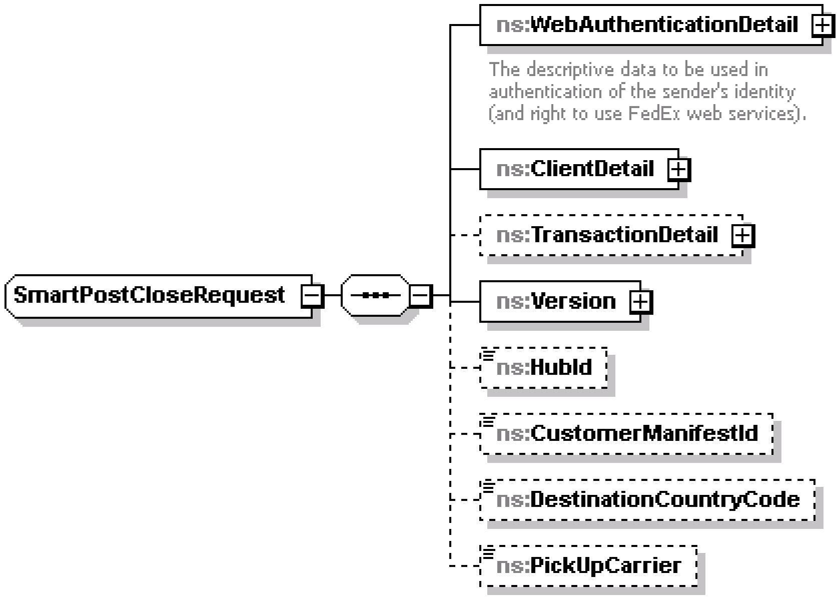
used by element SmartPostCloseReply

source

```
<xs:complexType name="SmartPostCloseReply">
  <xs:sequence>
    <xs:element name="HighestSeverity" type="ns:NotificationSeverityType"/>
    <xs:element name="Notifications" type="ns:Notification" maxOccurs="unbounded"/>
    <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"/>
    <xs:element name="Version" type="ns:VersionId"/>
  </xs:sequence>
</xs:complexType>
```

complexType **SmartPostCloseRequest**

diagram


SmartPostCloseRequest
ns:WebAuthenticationDetail
The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services).
ns:ClientDetail
ns:TransactionDetail
ns:Version
ns:HubId
ns:CustomerManifestId
ns:DestinationCountryCode
ns:PickUpCarrier

namespace http://fedex.com/ws/close/v2

children **ns:WebAuthenticationDetail ns:ClientDetail ns:TransactionDetail ns:Version ns:HubId ns:CustomerManifestId ns:DestinationCountryCode ns:PickUpCarrier**

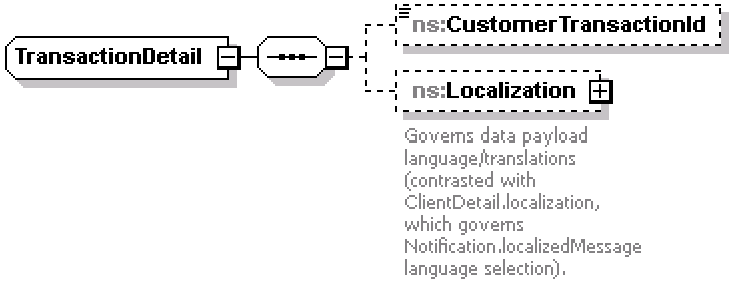
used by element SmartPostCloseRequest

source

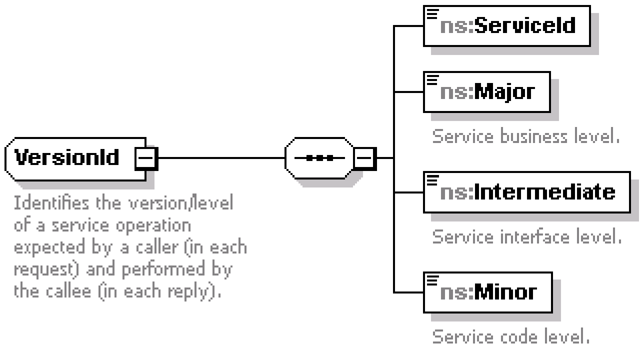
```
<xs:complexType name="SmartPostCloseRequest">
  <xs:sequence>
    <xs:element name="WebAuthenticationDetail" type="ns:WebAuthenticationDetail">
      <xs:annotation>
        <xs:documentation>The descriptive data to be used in authentication of the sender's identity
(and right to use FedEx web services).</xs:documentation>
      </xs:annotation>
```

	</xs:element> <xs:element name="ClientDetail" type="ns:ClientDetail"/> <xs:element name="TransactionDetail" type="ns:TransactionDetail" minOccurs="0"/> <xs:element name="Version" type="ns:VersionId"/> <xs:element name="HubId" type="xs:string" minOccurs="0"/> <xs:element name="CustomerManifestId" type="xs:string" minOccurs="0"/> <xs:element name="DestinationCountryCode" type="xs:string" minOccurs="0"/> <xs:element name="PickUpCarrier" type="ns:CarrierCodeType" minOccurs="0"/> </xs:sequence> </xs:complexType>

complexType **TransactionDetail**

diagram	 TransactionDetail ns:CustomerTransactionId ns:Localization Governs data payload language/translations (contrasted with ClientDetail.localization, which governs Notification.localizedMessage language selection).
namespace	http://fedex.com/ws/close/v2
children	**ns:CustomerTransactionId ns:Localization**
used by	elements **GroundCloseDocumentsReply/TransactionDetail GroundCloseReply/TransactionDetail GroundCloseReportsReprintReply/TransactionDetail GroundCloseReportsReprintRequest/TransactionDetail GroundCloseRequest/TransactionDetail GroundCloseWithDocumentsRequest/TransactionDetail ReprintGroundCloseDocumentsRequest/TransactionDetail SmartPostCloseReply/TransactionDetail SmartPostCloseRequest/TransactionDetail**
source	<xs:complexType name="TransactionDetail"> <xs:sequence> <xs:element name="CustomerTransactionId" type="xs:string" minOccurs="0"/> <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>

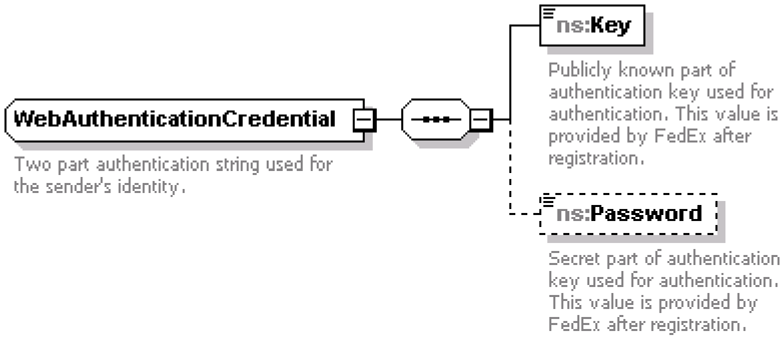
complexType **VersionId**

diagram	 VersionId Identifies the version/level of a service operation expected by a caller (in each request) and performed by the callee (in each reply). ns:ServiceId ns:Major Service business level. ns:Intermediate Service interface level. ns:Minor Service code level.
namespace	http://fedex.com/ws/close/v2

children	**ns:ServiceId ns:Major ns:Intermediate ns:Minor**
used by	elements **GroundCloseDocumentsReply/Version GroundCloseReply/Version GroundCloseReportsReprintReply/Version GroundCloseReportsReprintRequest/Version GroundCloseRequest/Version GroundCloseWithDocumentsRequest/Version ReprintGroundCloseDocumentsRequest/Version SmartPostCloseReply/Version SmartPostCloseRequest/Version**
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="clos"/>
	    <xs:element name="Major" type="xs:int" fixed="2">
	      <xs:annotation>
	        <xs:documentation>Service business level.</xs:documentation>
	      </xs:annotation>
	    </xs:element>
	    <xs:element name="Intermediate" type="xs:int" fixed="1">
	      <xs:annotation>
	        <xs:documentation>Service interface level.</xs:documentation>
	      </xs:annotation>
	    </xs:element>
	    <xs:element name="Minor" type="xs:int" fixed="0">
	      <xs:annotation>
	        <xs:documentation>Service code level.</xs:documentation>
	      </xs:annotation>
	    </xs:element>
	  </xs:sequence>
	</xs:complexType>

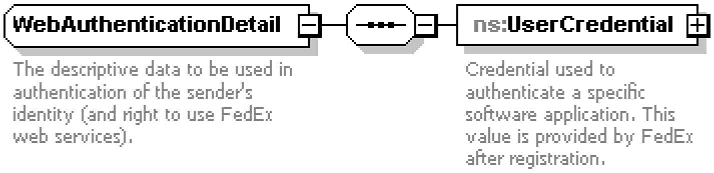
complexType **WebAuthenticationCredential**

diagram	 WebAuthenticationCredential Two part authentication string used for the sender's identity. ns:Key Publicly known part of authentication key used for authentication. This value is provided by FedEx after registration. ns:Password Secret part of authentication key used for authentication. This value is provided by FedEx after registration.
namespace	http://fedex.com/ws/close/v2
children	**ns:Key ns:Password**
used by	element **WebAuthenticationDetail/UserCredential**
annotation	documentation Two part authentication string used for the sender's identity.

source	<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">

```
    <xs:annotation>
      <xs:documentation>Publicly known part of authentication key used for authentication. This value is provided by FedEx after registration.</xs:documentation>
      <xs:appinfo>
        <xs:MaxLength>16</xs:MaxLength>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
  <xs:element name="Password" type="xs:string" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Secret part of authentication key used for authentication. This value is provided by FedEx after registration.</xs:documentation>
      <xs:appinfo>
        <xs:MaxLength>25</xs:MaxLength>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
 </xs:sequence>
</xs:complexType>
```

complexType **WebAuthenticationDetail**

diagram	 WebAuthenticationDetail The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services). ns:UserCredential Credential used to authenticate a specific software application. This value is provided by FedEx after registration.
namespace	http://fedex.com/ws/close/v2
children	**ns:UserCredential**
used by	elements **GroundCloseReportsReprintRequest/WebAuthenticationDetail** **GroundCloseRequest/WebAuthenticationDetail** **GroundCloseWithDocumentsRequest/WebAuthenticationDetail** **ReprintGroundCloseDocumentsRequest/WebAuthenticationDetail** **SmartPostCloseRequest/WebAuthenticationDetail**
annotation	documentation The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services).

source

```
<xs:complexType name="WebAuthenticationDetail">
 <xs:annotation>
  <xs:documentation>The descriptive data to be used in authentication of the sender's identity (and right to use FedEx web services).</xs:documentation>
 </xs:annotation>
 <xs:sequence>
  <xs:element name="UserCredential" type="ns:WebAuthenticationCredential">
   <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 **CarrierCodeType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **SmartPostCloseRequest/PickUpCarrier**
facets	Kind Value annotation

	enumeration FDXC enumeration FDXE enumeration FDXG enumeration FXCC enumeration FXFR enumeration FXSP
annotation	documentation Identification of a FedEx operating company (transportation).

source

```
<xs:simpleType name="CarrierCodeType">
  <xs:annotation>
    <xs:documentation>Identification of a FedEx operating company
(transportation).</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="FDXC"/>
    <xs:enumeration value="FDXE"/>
    <xs:enumeration value="FDXG"/>
    <xs:enumeration value="FXCC"/>
    <xs:enumeration value="FXFR"/>
    <xs:enumeration value="FXSP"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType **CloseDocumentType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	elements **CloseDocumentSpecification/CloseDocumentTypes CloseDocument/Type**
facets	Kind Value annotation enumeration COD_REPORT enumeration MANIFEST enumeration MULTIWEIGHT_REPORT enumeration OP_950

source

```
<xs:simpleType name="CloseDocumentType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="COD_REPORT"/>
    <xs:enumeration value="MANIFEST"/>
    <xs:enumeration value="MULTIWEIGHT_REPORT"/>
    <xs:enumeration value="OP_950"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType **CloseReportType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **GroundCloseReportsReprintRequest/CloseReportType**
facets	Kind Value annotation enumeration ALL enumeration COD enumeration HAZMAT enumeration MANIFEST enumeration MULTIWEIGHT

source

```
<xs:simpleType name="CloseReportType">
```

	<xs:restriction base="xs:string"> <xs:enumeration value="ALL"/> <xs:enumeration value="COD"/> <xs:enumeration value="HAZMAT"/> <xs:enumeration value="MANIFEST"/> <xs:enumeration value="MULTIWEIGHT"/> </xs:restriction> </xs:simpleType>

simpleType **CustomerImageUsageType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **CustomerImageUsage/Type**
facets	Kind Value annotation enumeration LETTER_HEAD enumeration SIGNATURE
source	<xs:simpleType name="CustomerImageUsageType"> <xs:restriction base="xs:string"> <xs:enumeration value="LETTER_HEAD"/> <xs:enumeration value="SIGNATURE"/> </xs:restriction> </xs:simpleType>

simpleType **EMailNotificationRecipientType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **ShippingDocumentEMailRecipient/RecipientType**
facets	Kind Value annotation enumeration BROKER enumeration OTHER enumeration RECIPIENT enumeration SHIPPER enumeration THIRD_PARTY
annotation	documentation Identifies the set of valid email notification recipient types. For SHIPPER, RECIPIENT and BROKER the email address asssociated with their definitions will be used, any email address sent with the email notification for these three email notification recipient types will be ignored.
source	<xs:simpleType name="EMailNotificationRecipientType"> <xs:annotation> <xs:documentation>Identifies the set of valid email notification recipient types. For SHIPPER, RECIPIENT and BROKER the email address asssociated with their definitions will be used, any email address sent with the email notification for these three email notification recipient types will be ignored.</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:enumeration value="BROKER"/> <xs:enumeration value="OTHER"/> <xs:enumeration value="RECIPIENT"/> <xs:enumeration value="SHIPPER"/> <xs:enumeration value="THIRD_PARTY"/> </xs:restriction> </xs:simpleType>

simpleType **ImageId**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **CustomerImageUsage/Id**
facets	Kind Value annotation enumeration IMAGE_1 enumeration IMAGE_2 enumeration IMAGE_3 enumeration IMAGE_4 enumeration IMAGE_5

source	`<xs:simpleType name="ImageId">`
	`  <xs:restriction base="xs:string">`
	`    <xs:enumeration value="IMAGE_1"/>`
	`    <xs:enumeration value="IMAGE_2"/>`
	`    <xs:enumeration value="IMAGE_3"/>`
	`    <xs:enumeration value="IMAGE_4"/>`
	`    <xs:enumeration value="IMAGE_5"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

simpleType **InternalImageType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **CustomerImageUsage/InternalImageType**
facets	Kind Value annotation enumeration LETTER_HEAD enumeration SIGNATURE

source	`<xs:simpleType name="InternalImageType">`
	`  <xs:restriction base="xs:string">`
	`    <xs:enumeration value="LETTER_HEAD"/>`
	`    <xs:enumeration value="SIGNATURE"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

simpleType **LinearUnits**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **LinearMeasure/Units**
facets	Kind Value annotation enumeration CM enumeration IN
annotation	documentation CM = centimeters, IN = inches

source	`<xs:simpleType name="LinearUnits">`
	`  <xs:annotation>`
	`    <xs:documentation>CM = centimeters, IN = inches</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`    <xs:enumeration value="CM"/>`
	`    <xs:enumeration value="IN"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

simpleType **NotificationSeverityType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	elements **GroundCloseDocumentsReply/HighestSeverity GroundCloseReply/HighestSeverity GroundCloseReportsReprintReply/HighestSeverity SmartPostCloseReply/HighestSeverity Notification/Severity**
facets	

Kind	Value	annotation
enumeration	ERROR	
enumeration	FAILURE	
enumeration	NOTE	
enumeration	SUCCESS	
enumeration	WARNING	

source

```
<xs:simpleType name="NotificationSeverityType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ERROR"/>
    <xs:enumeration value="FAILURE"/>
    <xs:enumeration value="NOTE"/>
    <xs:enumeration value="SUCCESS"/>
    <xs:enumeration value="WARNING"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType **ReprintGroundCloseDocumentsOptionType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **ReprintGroundCloseDocumentsRequest/ReprintOption**
facets	

Kind	Value	annotation
enumeration	BY_SHIP_DATE	
enumeration	BY_TRACKING_NUMBER	

annotation	documentation Identifies the requested options to reprinting Ground Close Documents

source

```
<xs:simpleType name="ReprintGroundCloseDocumentsOptionType">
  <xs:annotation>
    <xs:documentation>Identifies the requested options to reprinting Ground Close
Documents</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="BY_SHIP_DATE"/>
    <xs:enumeration value="BY_TRACKING_NUMBER"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType **ShippingDocumentDispositionType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	elements **ShippingDocumentDispositionDetail/DispositionType CloseDocument/ShippingDocumentDisposition**
facets	

Kind	Value	annotation
enumeration	CONFIRMED	
enumeration	DEFERRED	
enumeration	EMAILED	
enumeration	QUEUED	

	enumeration RETURNED enumeration STORED
annotation	documentation Specifies how to return a shipping document to the caller.

source

```
<xs:simpleType name="ShippingDocumentDispositionType">
  <xs:annotation>
    <xs:documentation>Specifies how to return a shipping document to the
caller.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="CONFIRMED"/>
    <xs:enumeration value="DEFERRED"/>
    <xs:enumeration value="EMAILED"/>
    <xs:enumeration value="QUEUED"/>
    <xs:enumeration value="RETURNED"/>
    <xs:enumeration value="STORED"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType **ShippingDocumentEMailGroupingType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **ShippingDocumentEMailDetail/Grouping**
facets	Kind Value annotation enumeration BY_RECIPIENT enumeration NONE
annotation	documentation Identifies the convention by which documents are to be grouped as e-mail attachments.

source

```
<xs:simpleType name="ShippingDocumentEMailGroupingType">
  <xs:annotation>
    <xs:documentation>Identifies the convention by which documents are to be grouped as e-mail
attachments.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="BY_RECIPIENT"/>
    <xs:enumeration value="NONE"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType **ShippingDocumentGroupingType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **ShippingDocumentDispositionDetail/Grouping**
facets	Kind Value annotation enumeration CONSOLIDATED_BY_DOCUMENT_TYPE enumeration INDIVIDUAL
annotation	documentation Specifies how to organize all shipping documents of the same type.

source

```
<xs:simpleType name="ShippingDocumentGroupingType">
  <xs:annotation>
    <xs:documentation>Specifies how to organize all shipping documents of the same
type.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="CONSOLIDATED_BY_DOCUMENT_TYPE"/>
```

	<xs:enumeration value="INDIVIDUAL"/> </xs:restriction> </xs:simpleType>

simpleType **ShippingDocumentImageType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **CloseDocumentFormat/ImageType**
facets	Kind Value annotation enumeration DIB enumeration DOC enumeration DPL enumeration EPL2 enumeration GIF enumeration PDF enumeration PNG enumeration RTF enumeration TEXT enumeration ZPLII
annotation	documentation Specifies the image format used for a shipping document.
source	<xs:simpleType name="ShippingDocumentImageType"> <xs:annotation> <xs:documentation>Specifies the image format used for a shipping document.</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:enumeration value="DIB"/> <xs:enumeration value="DOC"/> <xs:enumeration value="DPL"/> <xs:enumeration value="EPL2"/> <xs:enumeration value="GIF"/> <xs:enumeration value="PDF"/> <xs:enumeration value="PNG"/> <xs:enumeration value="RTF"/> <xs:enumeration value="TEXT"/> <xs:enumeration value="ZPLII"/> </xs:restriction> </xs:simpleType>

simpleType **ShippingDocumentNamingType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **ShippingDocumentStorageDetail/FileNaming**
facets	Kind Value annotation enumeration FAST enumeration LEGACY_FXRS
annotation	documentation Identifies the convention by which file names are constructed for STORED or DEFERRED documents.
source	<xs:simpleType name="ShippingDocumentNamingType"> <xs:annotation> <xs:documentation>Identifies the convention by which file names are constructed for STORED or DEFERRED documents.</xs:documentation>

```
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="FAST"/>
    <xs:enumeration value="LEGACY_FXRS"/>
  </xs:restriction>
</xs:simpleType>
```

simpleType **ShippingDocumentStockType**

namespace	http://fedex.com/ws/close/v2
type	restriction of **xs:string**
used by	element **CloseDocumentFormat/StockType**

facets

Kind	Value	annotation
enumeration	OP_900_LG	
enumeration	OP_900_LL	
enumeration	OP_950	
enumeration	PAPER_4X6	
enumeration	PAPER_LETTER	
enumeration	STOCK_4X6	
enumeration	STOCK_4X6.75_LEADING_DOC_TAB	
enumeration	STOCK_4X6.75_TRAILING_DOC_TAB	
enumeration	STOCK_4X8	
enumeration	STOCK_4X9_LEADING_DOC_TAB	
enumeration	STOCK_4X9_TRAILING_DOC_TAB	

annotation

documentation
Specifies the type of paper (stock) on which a document will be printed.

source

```
<xs:simpleType name="ShippingDocumentStockType">
  <xs:annotation>
    <xs:documentation>Specifies the type of paper (stock) on which a document will be printed.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="OP_900_LG"/>
    <xs:enumeration value="OP_900_LL"/>
    <xs:enumeration value="OP_950"/>
    <xs:enumeration value="PAPER_4X6"/>
    <xs:enumeration value="PAPER_LETTER"/>
    <xs:enumeration value="STOCK_4X6"/>
    <xs:enumeration value="STOCK_4X6.75_LEADING_DOC_TAB"/>
    <xs:enumeration value="STOCK_4X6.75_TRAILING_DOC_TAB"/>
    <xs:enumeration value="STOCK_4X8"/>
    <xs:enumeration value="STOCK_4X9_LEADING_DOC_TAB"/>
    <xs:enumeration value="STOCK_4X9_TRAILING_DOC_TAB"/>
  </xs:restriction>
</xs:simpleType>
```