

Address Validation XML Developers Guide

December 31, 2012



Notice

In all communications with UPS concerning this document, please refer to the document date located on the cover.

Copyright

The use, disclosure, reproduction, modification, transfer, or transmittal of this work for any purpose in any form or by any means without the written permission of United Parcel Service is strictly prohibited.

© Copyright 2012 United Parcel Service of America, Inc. All Rights Reserved.

Trademarks

Some of the UPS corporate applications use U.S. city, state, and postal code information obtained by United Parcel Service of America, Inc. under a non-exclusive license from the United States Postal Service.

Table of Contents

1.1 Important Information.....	5
1.2 Welcome to the UPS API Developer's Guides	6
1.2.1 Release Features.....	6
1.2.2 How to Use this Guide.....	7
1.3 Business Processes and Rules.....	8
1.4 Required Steps for Integrating	8
1.5 Planning Your Applications for XML	9
1.5.1 UPS Developer API Applications.....	9
1.5.2 Licensing the UPS Developer APIs.....	10
1.5.2.1 Branding Requirements	11
1.5.3 Developing Client Applications for Developer APIs	11
1.5.3.1 Microsoft Visual Studio.....	11
1.5.3.2 Java Standard Edition and Enterprise Edition	11
1.5.3.3 Open Source	11
1.5.4 Getting Technical Support	12
1.5.5 Testing and Deploying Applications	15
1.5.6 Keeping Up-to-Date.....	15
1.6 UPS Developer API Technologies for XML	15
1.6.1 Hypertext Transfer Protocol and Secure Sockets Layer 3.....	15
1.6.2 Working with HTTP and SSL in Application Programs	17
1.6.2.1 Microsoft Visual Studio.....	18
1.6.2.2 Java Standard Edition and Enterprise Edition	19
1.6.2.3 Open Source	19
1.6.3 SSL Certificate Changes and Renewals:.....	20
1.6.4 Extensible Markup Language (XML).....	22
1.6.5 Working with XML in Application Programs	24
1.6.5.1 Microsoft Visual Studio.....	24
1.6.5.2 Java Standard Edition and Enterprise Edition	26
1.6.5.3 Open Source	27
1.7 Understanding XML Address Validation Services.....	29
1.7.1 Displaying the UPS Address Validation Notice	30
1.8 The Address Validation XML Reference	31
1.8.1 AV Request	36
1.8.2 AV Response	39
1.9 Customer Integration Environment.....	44
1.9.1 Address Validation.....	44
1.9.2 System Availability.....	44
1.9.3 Server Availability Check.....	44

Appendices

Appendix A - Accessing Tech Support (US Site)	45
Appendix B – Address Validation Error Codes.....	49
Appendix C - Country Codes.....	51
Appendix D - Frequently Asked Questions: Address Validation.....	58

1.1 Important Information

UPS Developer APIs

Your development of an application using the UPS Web Service APIs are governed by the UPS Technology Agreement or UPS Customer Technology Agreement you entered into with UPS. The following are key legal requirements from these agreements for the UPS Web Service APIs. For more information on all requirements for the UPS Web Service APIs, please refer to the UPS Technology Agreement or the Customer Technology Agreement.

Key Legal Requirements for UPS Developer APIs

Permitted Territories

This document can only be used in the countries listed in Exhibit C of the UPS Technology Agreement or UPS Customer Technology Agreement.

Use

The application must not be designed to allow distribution of information received through the UPS Web Service APIs to third parties, other than to persons having a bona fide interest in such information (e.g., the shipper, receiver or the third party payer).

Consent to Use of UPS Mark

- All screens or forms generated by your application including information received through the UPS Web Service APIs must include (1) the UPS Mark positioned in reasonable proximity to the Information and of an appropriate size to readily identify the source of the Information as UPS and (2) the following language at the bottom of every screen that displays the UPS Mark: "UPS, the UPS brand mark, and the Color Brown are trademarks of United Parcel Service of America, Inc. All Rights Reserved". Except as set forth in the preceding sentence, you have no right to use the UPS Mark without the prior written approval of UPS.
- You shall not use the UPS Mark in association with any third party trademarks in a manner that might suggest co-branding or otherwise create potential confusion as to source or sponsorship of the application, or ownership of the UPS Mark.
- The UPS Mark shall be used only as provided by UPS electronically or in hard copy form. The UPS Mark may not be altered in any manner, including proportions, colors, elements, etc., or animated, morphed or otherwise distorted in perspective or dimensional appearance.
- The UPS Mark may not be combined with any other symbols, including words, logos, icons, graphics, photos, slogans, numbers or other design elements. A minimum amount of empty space must surround the UPS Mark separating it from any other object, such as type, photography, borders, edges, etc. The required area of empty space around the UPS Mark must be $1/3x$, where x equals the height of the UPS Mark.

Copyright and Proprietary Notice

In your application and any POD Letters you prepare you must include a prominent reproduction of UPS's copyright and proprietary notices in a form and format specified by UPS (See Copyright Section of this document).

Display of Information

The application must not display information concerning any other provider of shipping services or such other shipping services on any page, whether comprising one or more frames, displaying information your application receives from the UPS Web Service APIs. Your application must present all data within each field received through the UPS Web Service APIs without amendment, deletion or modification of any type.

1.2 Welcome to the UPS API Developer's Guides

Welcome to the UPS API Developer's Guides. This guide provides the information you need to begin using UPS Developer APIs.

UPS Developer APIs offer a fast and convenient way to access UPS service information using the Internet. With these Developer APIs, UPS lets you easily incorporate UPS technology in your own applications or your own web site. Your users—running your applications or visiting your web site—can have up-to-the-minute access to UPS services.

1.2.1 Release Features

Release	New Features
Jan 2013	Technical Support Email Template Improvements - Optionally collect the customer's UPS Account Number, User ID and Access Key - Capture if UPS Ready Provider (Y/N); Default field selection to "N" - Update email application content on what the types of attachment is not allowed (EXE) for a logged-in user. - Include the Stage of Development (Pre-Integration, Integration, and Post-Integration) in the email subject line (end of existing text); make Stage of Development a mandatory field. - Under Attach File inform the requester that XML Request and Response files should be included whenever possible. UPS Developer Kit Community - New System Status Page - Customers can now register for the UPS Developer Kit Community and subscribe to the 'System Status Page' for proactive email alerts when system status changes occur

	(Note: UPS Developer Kit Community can be accessed from the UPS Developer Resource Center) New User Guide and Quick Reference Guide Available at UPS Developer Resource Center • There are now two new customer reference materials never before available. The first is a UPS Developer Kit User Guide, which takes the developer through the step by step process of gaining access to API documentation, testing/production environments, and available tech support resources. The second is a UPS Developer Kit Quick Reference Guide that summarizes at a very high level gaining access to the UPS Developer Kit and tech support resources. • New User Guide (comprehensive access guide) • New Quick Reference Guide (high level quick reference guide)
--	--

1.2.2 How to Use this Guide

If you are an experienced developer, you can begin developing applications quickly after reviewing "Required Steps for Integrating."

If you would like a more step-by-step guide to developing and deploying the Developer APIs, "Planning Your Applications" provides advice and describes options for developing and deploying applications and web sites that use UPS Developer APIs.

The "UPS Developer API Technologies" section explains key technologies on which the Developer APIs rely. That section also includes hints for using those technologies in various software development environments.

If you would like to learn more about what the UPS API covered in this guide can do for your applications, refer to the section on understanding the UPS API Services in this guide.

A complete technical reference to the Developer API covered in this guide is found in the API Reference section with details for the programming interfaces.

Additional material, including reference tables and lists, may be found in the appendices.

1.3 Business Processes and Rules

- UPS restricts the usage of the Address Validation API for only packages manifested, tendered, and delivered by UPS.
- Any customers/developers abusing the Address Validation API or data mining the API will have their access revoked.
- To obtain testing and/or production access to the Address Validation API you need to go through the Request Access Key process from the Developer Resource Center on UPS.com. During the process the user is required to either create a new account number or to add an existing account number from their profile. Access Keys to use the API will not be created unless an account number is provided.
- Account numbers can only be added to UPS.com profiles by providing the following additional information for authentication:
 - Account Number
 - Account Country Code
 - Invoice Level Control ID
 - Plan Level Control ID
 - Amount Due on Invoice
 - Date of Invoice

1.4 Required Steps for Integrating

The required steps for integrating the UPS APIs are listed here for UPS XML Services.

UPS XML Services

1. Review the UPS Technology Agreement available at www.ups.com. This agreement requires that you follow certain procedures and practices in using UPS Developer APIs.
2. Develop applications that make use of standard HTTP communications protocols, SSL security, and XML-based document representations.
3. Test your application and/or web site using the designated UPS staging environment.
4. If you are a UPS Ready developer, review your application with UPS.
5. Obtain UPS Label Certification for your application. (If you are a new customer or if you haven't previously certified. Please see the "UPS Label Certification" section for further details.).
6. Deploy your application for your customers.
7. Ensure the UPS Shipper Account number is added to the user's profile. This can be done at myups.com.

1.5 Planning Your Applications for XML

Planning is a key part of any successful development activity, and UPS Developer API applications are no exception. This section helps that planning by describing the important activities of any UPS Developer API application project. It provides an overview of the steps required to develop applications, and it describes key factors and choices necessary to deploy those applications.

1.5.1 UPS Developer API Applications

Many different types of applications can take advantage of UPS Developer APIs. Those application types include dedicated desktop applications, databases, web applications, and documents. The following figures illustrate some of the possibilities for applications using UPS Developer APIs. The only essential requirement for all of these applications is that they must have access to the Internet.



Figure 1: Dedicated applications that users run on their desktops can access UPS Developer APIs.

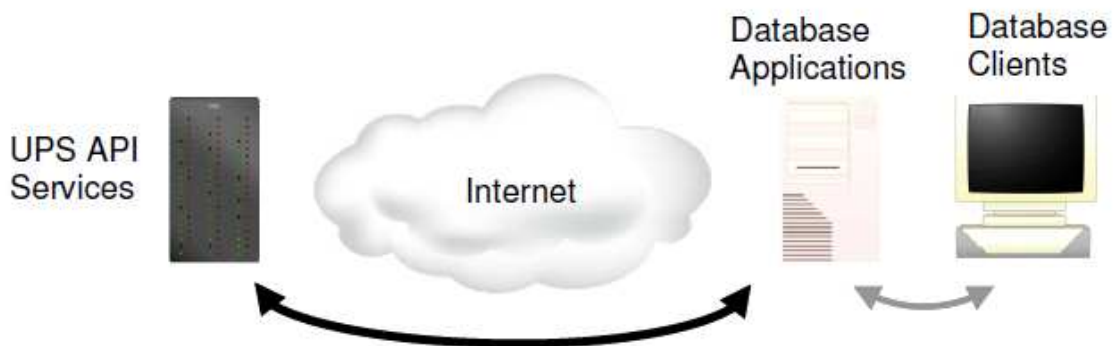


Figure 2: Database applications can access UPS Developer APIs and return information to their clients

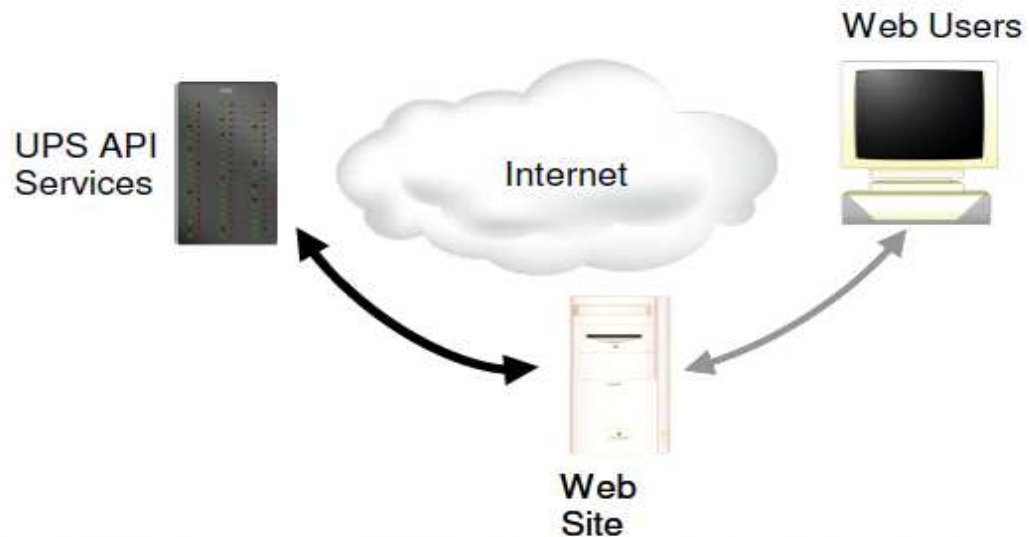


Figure 3: Web sites can access UPS Developer APIs and return information to users' web browsers.



Figure 4: Non-traditional applications such as Microsoft Office or Adobe Acrobat documents can use UPS Developer APIs to automatically update their content.

The UPS Developer APIs can be implemented with all of these types of applications and many others. Virtually any software that needs instant, up-to-date access to UPS services can take advantage of UPS Developer APIs.

1.5.2 Licensing the UPS Developer APIs

As part of the UPS Technology Agreement, users of the APIs have certain obligations that are spelled out within the service agreement and its exhibits. Regardless of the manner in which the UPS Developer APIs are integrated into your specific e-commerce web site or enterprise application, you must adhere to appropriate usage requirements.

1.5.2.1 Branding Requirements

UPS should receive attribution and branding in all applications (including websites and software applications) that use the Developer APIs. No End User, Third Party Developer or Access User should be permitted to use the Developer APIs without providing branded recognition to UPS. Your use of the UPS logo can in no way imply endorsement, sponsorship or certification of your ecommerce web site or enterprise application by UPS. You are not allowed to use or alter the information returned by the UPS Developer APIs in a way that misrepresents the information or the functionality of the service.

1.5.3 Developing Client Applications for Developer APIs

When you develop software that uses the UPS Developer APIs, you are building a client application. Because the UPS Developer APIs rely on standard Internet based technology, you can develop those applications using a wide variety of software development platforms, including Microsoft's Visual Studio, the Java Standard Edition and Enterprise Edition distributions, and many open source projects. This section introduces important technologies available in each of these development environments. Later sections of this document include more details on using each platform.

1.5.3.1 Microsoft Visual Studio

Developers using the Microsoft Visual Studio environment can rely on the Microsoft XML Core Services (MSXML) for interacting with UPS Developer APIs. Those services include functions to help applications create requests for and interpret responses, and they include functions to manage the communications between applications and UPS.

1.5.3.2 Java Standard Edition and Enterprise Edition

Java developers can find all the classes they need for UPS Developer APIs in the Java Standard Edition and Enterprise Edition distributions. The URL class in the java.net package provides the functions required for communication with UPS servers, and the Java API for XML Processing (JAXP) services let programs create requests and interpret responses from UPS.

1.5.3.3 Open Source

There are a number of open source efforts that include technology useful for creating Developer API client applications; open source developers may find two particular projects to be especially helpful. The libcurl project is a library of functions that can manage the communications with UPS servers. For creating requests and interpreting responses, the xerces project provides essential functions in a variety of software languages.

1.5.4 Getting Technical Support

Technical Support (US)

There are four channels for obtaining support for the UPS Developer Kit (UDK) APIs all of which are accessed through the **UPS Developer Resource Center** at the following link:

<http://www.ups.com/content/us/en/resources/techsupport/developercenter.html?WT.svl=SubNav>

Or by following these steps:

1. Go to www.ups.com.
 2. Mouse over the Support tab and select Technology Support
 3. In the left navigation panel of the page select the link “Developer Resource Center”.
- Email Technical Support – available in the right hand column on the **UPS Developer Resource Center** web page. This is *the* support channel for technical support and questions regarding API integration. It is accessed by logging in to My UPS and attaching your XML Request/Response files and any other pertinent information about your integration [please see Appendix A for detailed instructions and screen shots]. UPS Email Technical Support can only support the XML Request/Response pair. They cannot consult customers and developers on how to integrate the XML into their internal or purchased software.
 - UPS Developer Kit Community – available in the right hand column of the **UPS Developer Resource Center** web page. This is a community forum for developers to answer one another’s questions and share information therefore the response times may be slower. If you have an immediate need for technical support contact UPS Email Tech Support above and they will respond within 4 business hours (see hours below). The remainder of the site contains online versions of all API developer guides and FAQs.
 - UPS Developer Kit Knowledge Base – available in the right hand column of the **UPS Developer Resource Center** web page. This is a self-service support PDF document that can be opened or downloaded. It contains over 240 frequently asked questions about every mode and service supported by the UDK APIs. It’s also available in an online version at the UPS Developer Kit Community.
 - General Question Phone Support (U.S. only) – available in the right hand column of the **UPS Developer Resource Center** web page. Phone support is the *least comprehensive* support mechanism for the developer. This channel of support addresses questions about the user interface at ups.com and how to gain access to the APIs and navigate the Access Key request process. They do not provide technical support for XML Request/Response issues and questions. Those types of questions must go through Email Technical Support.

Getting Started with Email Technical Support

1. Got to ups.com and log in to My UPS.
2. Hover over the “Support” tab and select “Technology Support”.
3. On the Technology Support page select the Developer Resource Center link from the left navigation.
4. On the Developer Resource Center page select the “Email UPS” link in the right hand column.
5. Enter all relevant information including name, enter email address, select Support Category “Technical Support”, and select Support Topic “Developer Resource” [which identifies UPS Developer Kit APIs tech support]. Click next.
6. Scroll down to the middle of the email form and complete the fields “Your Telephone”, “Stage of Development”, “Developer Resource” which is which API you are integrating, “Attach File” where you attach your XML Request/Response Pair, and any pertinent description of the issues in the “What is your question or comment?” field.
7. Select “Send Email” button.
8. A response is provided that explains someone from UPS will contact you. Please base the response time on the detailed information listed below about tech support hours of operation and response times.

(For screen shots of this process please see Appendix A at the end of this guide.)

1st Level Email Technical Support Business Hours and Response Times:

- a. Hours of operation for 1st Level Email Tech Support: 7:30am to 9pm EST Monday through Friday and 9am to 6pm EST Saturday and Sunday. The desk is closed on UPS Holidays.
- b. 1st Level Email Tech Support email response time is planned to be within 4 business hours of the initial request. Business hours are defined as hours of operation of the tech support desks when service technicians are working.

This means if you submit an email tech support request after business hours the 4 hour response clock will not begin until the following morning at 7:30am EST M-F and 9am EST Saturday and Sunday (UPS Holidays excluded).

PLEASE NOTE:

If you are sent an email from 1st Level Email Tech Support stating your case log # has been escalated, the 24 hour planned response clock starts from the time you receive the email from 1st Level Email Tech Support (provided the Escalation Support desk is open). If not, then the clock will begin when the Escalation Support desk opens.

Escalation Support Business Hours and Response Times:

- c. Hours of operation: 8am to 8pm EST Monday through Friday. The desk is closed weekends and on UPS Holidays.

- d. Escalation Support desk response time is planned to be within 24 hours during normal hours of operation.

This means if your case log is escalated at 8:30pm on Friday, you will not receive an email response from the Escalation Support desk at the latest until Monday at 8pm (Saturdays, Sundays, and Holidays are excluded). Response times are from the time the case log was escalated plus 24 hours which do NOT include the weekends or UPS holidays.

There may be some scenarios when the tech support team is meeting planned response times but it appears as if it's taking too long. Below is a brief example where the planned response time is met but nothing happened over the weekend.

Example:

Customer submits tech support email at 7am on Friday. The clock for 4 hour response will not begin until 7:30am EST for the 1st level tech support team. They respond within 4 hours at 11:30am EST that same Friday stating the case log has been escalated and providing the case log #.

The escalation desk receives the case log at 11:30am EST. They begin work on evaluating the problem and have planned to either update the customer within 24 hours or solve the problem within 24 hours. Either way, the customer will hear from the escalation desk within 24 hours with some type of disposition on their case log. The escalation desk clock begins when they receive a case log from 1st level tech support during business hours, M-F 8am to 8pm EST. In this case, the clock begins at 11:30am EST if the escalation desk does not have an answer by 8pm EST on Friday the desk has until 11:30am EST the following Monday to either provide a resolution or a status update to the customer on the case log. The Escalation Desk continues work on the case log on Monday beginning at 8am EST. They receive resolution at 1030am EST Monday and send the resolution to the customer. The Escalation Desk has met their 24 hour planned response time.

Technical Support (Non-US)

For Non-US countries supported by the UPS Developer Kit follow the instructions in Appendix A. However, to get to the UPS Developer Resource Center pages please complete the following steps.

1. Log in to My UPS for the country from which you downloaded the developer guides.
2. Select the Support tab and in the drop-down select Technology Support.
3. Select from the left navigation the "UPS Developer Resource Center" link.
4. Continue with email form as described above.

Please also note that email response times for non-US requests vary. Translations and escalations may delay the process. If a quicker response is required, please access the US web site directly and complete an email technical support form from the US web site.

1.5.5 Testing and Deploying Applications

UPS maintains a special environment to support testing and staging of applications that rely on UPS Developer APIs. This environment is called the Customer Integration Environment (CIE) and allows developers test and debug their applications by simulating transactions with UPS. The CIE site responds to requests just like the UPS production environment; however, it does not initiate actual UPS business services.

For example, if you send a shipping request to the UPS production site, a UPS driver may show up at your location expecting to pick up a package (and expecting payment for the service.) Sending the shipping request to CIE will avoid this problem.

1.5.6 Keeping Up-to-Date

As UPS adds new services and features, Developer APIs will evolve, offering more features and service benefits. Once you register to use UPS Developer APIs, UPS will notify you by e-mail of updates and changes to the Developer APIs. It is essential that an accurate e-mail address for your company be maintained. In addition, UPS recommends that you complete the secondary contact information to ensure that your organization receives the latest updates.

You should update your profile when changes or responsibilities for the UPS Developer APIs change within your company. You can also return to the UPS Support area of ups.com for the latest updated information about UPS Developer APIs.

1.6 UPS Developer API Technologies for XML

The foundation technologies for UPS Developer APIs are the same underlying technologies in use on the Internet today. They include the Hypertext Transfer Protocol (HTTP) with Secure Sockets Layer 3 (SSL3) security and the Extensible Markup Language (XML). Because these technologies are critical to the Internet, most software development tools provide many powerful features that make it easy for application programs to use them. This section provides a quick introduction to the technologies and to the tools that can support them.

1.6.1 Hypertext Transfer Protocol and Secure Sockets Layer 3

Today's Internet supports a wide variety of applications including email, instant messaging, web browsing, and voice communications. All of these applications have a well-defined set of rules that determines how they communicate. Those rules are known as *protocols*. As an Internet application themselves, the UPS Developer APIs also rely on protocols to set the rules for their communications. The primary protocol for UPS Developer APIs is the hypertext transfer protocol, or HTTP.

HTTP is a relatively simple protocol. An application that wants to use a Developer API acts as an HTTP client. It sends its request to UPS as part of an HTTP POST message. UPS servers reply to each request using an HTTP response message.

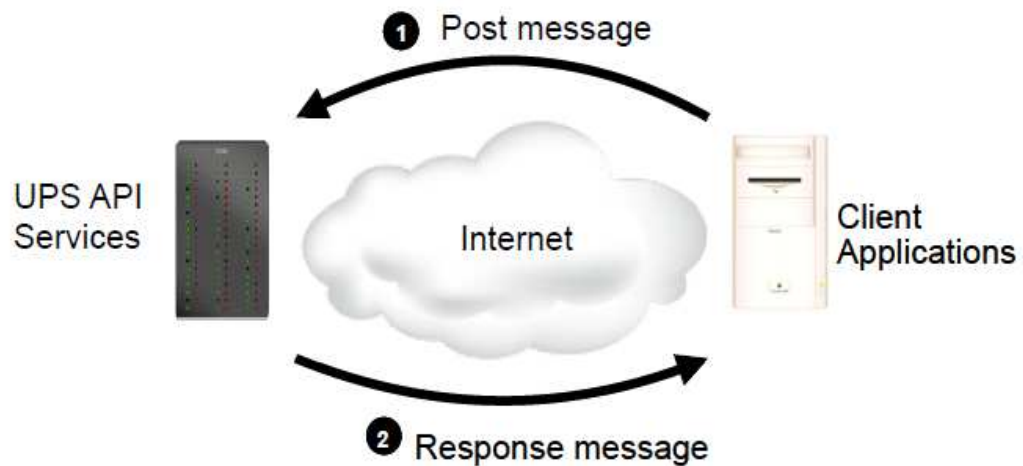


Figure 1. UPS Developer APIs accept requests from client applications in HTTP POST messages and reply to the requests with HTTP responses.

e POST message, client applications indicate the type of content the message contains using an HTTP Content-Type header. For UPS Developer APIs, that content type should be application/x-www-form-urlencoded. (Presently, Version 1.1 is supported)

In many cases the UPS Developer APIs exchange information that should be kept private. To protect confidential information, the Developer APIs rely on the Secure Sockets Layer 3 (SSL3) protocol in addition to HTTP. When two systems communicate using SSL, the protocol creates a secure channel between them, and it encrypts all information that they exchange using this channel. The SSL protocol that Developer APIs use is the same protocol used to secure millions of on-line purchases on the web.



Figure 2: SSL creates a secure channel across a network and protects confidential communications using that channel.

1.6.2 Working with HTTP and SSL in Application Programs

UPS Developer APIs are not the only Internet services that rely on the HTTP and SSL protocols. Both protocols, in fact, were originally developed for web browsing, and they provide the foundation for the majority of Internet applications. Because HTTP and SSL are so common, software development tools make using these protocols very simple.

Because HTTP and SSL are commonly used for web browsing, software development tools often rely on the same notation and abbreviations as standard web browsers. In particular, most tools identify a specific service (such as a UPS Developer API) using a Uniform Resource Locator (URL). URLs begin with an abbreviation of the communication protocol. For UPS Developer APIs that abbreviation will always be "https" to indicate HTTP and SSL. The protocol abbreviation is followed by a colon, two slashes, and the name of a server. Additional information about the specific service can follow the server name; it is written like the path to a file in a directory, with slashes separating individual folders. Figure 3 shows how a URL combines these individual components.

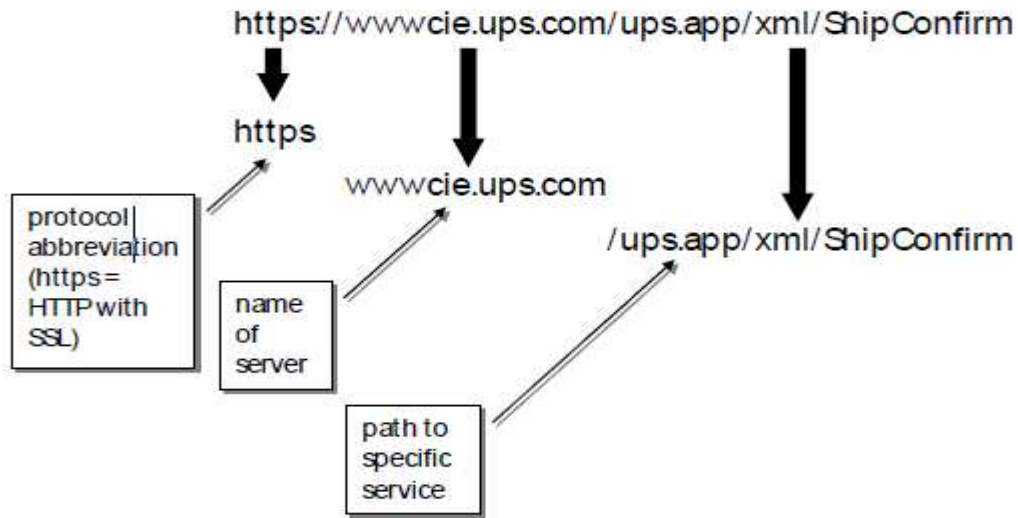


Figure 3. URLs identify communication protocols, servers, and specific services.

The subsections that follow provide brief introductions to using HTTP and SSL with various software development tools, including Microsoft Visual Studio, Java Standard and Enterprise Edition, and popular open source environments.

1.6.2.1 Microsoft Visual Studio

The Microsoft Visual Studio environment includes many tools, classes, and libraries that simplify the use of HTTP and SSL protocols. A very convenient set of tools are included in the Microsoft XML Core Services (MSXML). The following code fragment shows how a Visual Basic program can send a request and retrieve the response. To keep the example as simple as possible, no exception handling is included in the figure. Production software should, of course, appropriately handle all exceptions

```
' create the object that manages the communication
Dim oXMLHttp As XMLHTTP
Set oXMLHttp = New XMLHTTP

' prepare the HTTP POST request
oXMLHttp.open "POST", "https://www.server.com/path", False
oXMLHttp.setRequestHeader "Content-Type", _
    "application/x-www-form-urlencoded"

' send the request
oXMLHttp.send requestString

' server's response will be available in oXMLHttp.responseXML
```

Figure 4. Visual Basic programs can use features of the MSXML tools to send and receive messages using HTTP and SSL.

1.6.2.2 Java Standard Edition and Enterprise Edition

In a Java SE or EE environment, the URL class in the java.net package provides a convenient way to manage HTTP and SSL communications. Figure 5 contains a small code fragment that demonstrates the use of this class. As above, the fragment doesn't include exception handling that should be inherent in any production software.

```
import java.io.*;
import java.net.*;

URL url = new URL("https://www.server.com/path");
URLConnection conn = (URLConnection) url.openConnection();
conn.setRequestMethod("POST");
conn.setRequestProperty("Content-Type",
    "application/x-www-form-urlencoded");
conn.setDoOutput(true);
PrintWriter out = new PrintWriter(conn.getOutputStream());
out.println(requestString);
out.close();
BufferedReader in = new BufferedReader(
    new InputStreamReader(conn.getInputStream()));
/* server response is available by reading the in object */
```

Figure 5. The URL class from the java.net package is a convenient way for Java programs to use HTTP and SSL

1.6.2.3 Open Source

The libcurl project is an open source project that includes support for using HTTP and SSL protocols. The fragment in Figure 6 shows how to set up libcurl to send a request using HTTP and SSL. It relies on a callback function to accept the server's response.

```
CURL *curl;
CURLcode res;

/* prepare to send the request */
curl = curl_easy_init();
curl_easy_setopt(curl, CURLOPT_URL, https://www.server.com/path");
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, requestString);
curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, fnCallback);

res = curl_easy_perform(curl);
```

Figure 6. The open source libcurl library simplifies the use of HTTP and SSL

1.6.3 SSL Certificate Changes and Renewals:

UPS uses Chained Digital Certificates to improve security for UPS servers. Chained Digital Certificates requires the use of SSL 3.0.

Renewals of UPS.com SSL Certificates are coordinated by the UPS Security Services every year. There is a concern that some UPS Developer Kit users who store UPS Certificates internally may have issues when SSL Certificates are renewed. It is recommended that UPS Developer Kit users not store UPS Certificates internally; however if a customer finds this necessary due to specific needs within their company, they may need to add renewed certificates to the their trust store.

There are a number of ways one can add a certificate to the application. One way to obtain the proper certificate is by placing a UPS Developer Kit URL which is shown below, <https://onlinetools.ups.com/ups.app/xml/AV> , for example, in a browser, connect to the URL, and double-click on the "lock" on the bottom right of the window (using Internet Explorer 7). After that the steps can be followed to install the certificate.

Also, in the event that the aforementioned solution does not work successfully for you we have been provided with the following certificate which should allow you to connect once it has been successfully added. You should be able to paste the following string into Notepad and save the file locally. Then, it will be necessary to have the client application reference the file so that it will know that this is the trusted certificate for making a connection to UPS.

This is the Verisign Class 3 Secure Server CA - G2 certificate that the client must trust:

-----BEGIN CERTIFICATE-----

```
MIIGLDCCBZWgAwIBAgIQbk/6s8XmacTRZ8mSq+hYxDANBgkqhkiG9w0BAQUFAD
CB
wTElMAkGA1UEBhMCVVMxZmFzAVBgNVBAoTDlZlcm1TaWduLCBJbmMuMTwwOg
YDVQQL
EzNDbGFzcyAzIFB1Ym9yYyBQcm1tYXJ5J5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5
IC0gRzIxOjA4BgNVBAsTMSHjKSAxOTk4IFZlcm1TaWduLCBJbmMuIC0gRm9yIGF1
dGhvcml6ZWQgdXNlIG9ubHkxHzAdBgNVBAsTF1Zlcm1TaWduIFRydXN0IE5ldHdv
cm5wHhcNMDkwMzI1MDAwMDAwWhcNMTkwMzI0MjM1OTU5WjCBTELMAkGA
1UEBhMC
VVMxZmFzAVBgNVBAoTDlZlcm1TaWduLCBJbmMuMR8wHQYDVQQLExZWZXJpU2l
nbiBU
cnVzdCBOZXRx3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2UgYXQgaHR0cHM6Ly9
3
d3cudmVyaXNpZ24uY29tL3JwYSAoYykwOTEvMC0GA1UEAxMmVmVyaVNpZ24gQ
2xh
c3MgMyBTZW51cmUgU2VydMvYIENBIC0gRzIwggEiMA0GCSqGSIb3DQEBAQUA
A4IB
DwAwggEKAoIBAQUdUVo9XOzcopkBj0pXVBXTatRlqltZxVY/iwDSMoJWzjOE3JPMu
7UNFBY6J1/raSrX4Po1Ox/IJUEU3QJ90qqBRVWHxYISJpZ6AjS+wlapFgsTPtBR/
```

RxUgKIKwaBLArlwH1/ZZzMtiVlxNSf8miKtUUTovStoOmOKJcrn892g8xB85essX
gfMMrQ/cYWIbEAsEHikYcV5iy0PevjG6cQIZTiapUdqMZGkD3pz9ff17Ybz8hHyI
XLTD+1fK0YS8f0AAZqLW+mjBS6PLlve8xt4+GaRCMBeztWwNsrUqHugffkwer/4
3RIRKyC6/qfPoU6wZ/WAqiuDLtKOVImOHikLAgMBAAGjggKpMIICpTA0BggrBgEF
BQcBAQQoMCYwJAYIKwYBBQUHMAGGGH0dHA6Ly9vY3NwLnZlcmlzaWduLm
NvbTAS
BgNVHRMBAf8ECDAGAQH/AgEAMHAGA1UdIARpMGcwZQYLYIZIAyB4RQEHF
wMwVjAo
BggrBgEFBQcCARYcaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL2NwczAqBggrBgEF
BQcCAjAeGhxodHRwczovL3d3dy52ZXJpc2lnbi5jb20vcnBhMDQGA1UdHwQtMCsw
KaAnoCWGI2h0dHA6Ly9jcmwudmVyaXNpZ24uY29tL3BjYTMtZzluY3JsMA4GA1Ud
DwEB/wQEAwIBBjBtBggrBgEFBQcBDARhMF+hXaBbMFkwVzBVFGlpbWFnZS9naW
Yw
ITAfMAcGBSsOAwIaBBSP5dMahqyNjmvDz4Bq1EgYlHsZLjAlFiNodHRwOi8vbG9n
by52ZXJpc2lnbi5jb20vdnNsb2dvLmdpZjApBgNVHREEIjAgpB4wHDEaMBGA1UE
AxMRQ2xhc3MzQ0EyMDQ4LTtNTIwHQYDVR0OBBYEFKXvCxHwEEDo0plkEiy
HOBX
LX1HMIHnBgNVHSMEdg8wgdyhgcekgcQwgExCzAJBgNVBAYTAiVTMRcwFQYD
VQK
Ew5WZXJpU2lnbiwgSW5jLjE8MDDoGA1UECXMzQ2xhc3MgMyBQdWJsaWMgUHJpb
WFy
eSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eSAiIEcyMTowOAYDVQQLZeoYykgMTk
5
OCBWZXJpU2lnbiwgSW5jLiAtIEZvciBhdXR0b3JpemVkIHVzZSBvbmx5MR8wHQYD
VQQLExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrghB92f4Hz6getxB5Z/uniTTGMA0G
CSqGSIlb3DQEBBQUAA4GBAGN0Lz1Tqi+X7CYRZhr+8d5BJxnSf9jBHPniOFY6H5Cu
OcUgdav4bC1nHynCIdcUiGNLsJsnY5H48KMBJLb7j+M9AgtvVP7UzNvWhb98IR5e
YhHB2QmcQrmy1KotmDojYMyimvFu6M+O0Ro8XhnF15s1sAljJOUFuNWl4+D6ufRf

-----END CERTIFICATE-----

Finally, for more information on installing the latest VeriSign CA Root Certificate, please click on the links below:

https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&id=SO7154&actp=search&viewlocale=en_US&searchid=1308235124970
<https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&actp=CROSSLINK&id=AR1553>
<https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&id=SO4785&actp=LIST>

1.6.4 Extensible Markup Language (XML)

The Extensible Markup Language (XML) is an international standard developed by the World Wide Web Consortium, the governing body for web standards and guidelines. XML provides a way to identify the structure of content within a document or, in the case of UPS Developer APIs, a message. Figure 7 shows how a simple XML message could describe a book.

```
<?xml version="1.0" encoding="UTF-8" ?>
<book>
  <title>
    HTTP Essentials: Protocols for Secure, Scaleable Web Sites
  </title>
  <author>
    <firstname>
      Stephen
    </firstname>
    <lastname>
      Thomas
    </lastname>
  </author>
  <publisher>
    John Wiley and Sons
  </publisher>
  <year>
    2001
  </year>
  <isbn>
    0-471-398233
  </isbn>
</book>
```

Figure 7: XML identifies the structure of documents, as in this document describing a book.

As the figure illustrates, XML is a text-based format. XML messages contain regular text, though that text follows specific rules that XML defines. XML distinguishes different parts of a message with a label known as a *tag*. Tags in the example include `<book>`, `<title>`, `<author>`, `<firstname>`, etc. A tag can indicate the start of information if it begins with an angle bracket (`<`), and a tag can indicate the end of information if it begins with an angle bracket and a slash (`</>`). A beginning tag, ending tag, and the information between the two make up an *element*. In this example the publisher element tells us that the publisher for the book is "John Wiley and Sons."

The figure also shows how an XML message follows a defined structure. Elements can be contained within other elements, as `<firstname>` and `<lastname>` are included within the

<author> element. This structure indicates that firstname and lastname are "children" of the author "parent."

Although XML elements can contain almost any text, there are two special characters that cannot appear within an element. Those characters are the less-than sign (<) and the ampersand (&). Elements that include these contents must replace the special characters with "<" or "&" respectively. The name of a large telecommunications company, for example, would appear as "AT&T" within an XML message.

Because XML is a text-based format, software has to take extra steps to use XML with binary data. For example, some UPS services return binary image data such as the image of a shipping label. To include this data in an XML message, UPS converts it from binary to text format using an algorithm known as *Base64 encoding*. The "The Base16, Base32, and Base64 Data Encodings" standard, also known as [RFC 3548](#), defines that algorithm.

Although XML messages consist of a series of text characters, most software development tools represent XML messages in a tree-like data structure. That representation clarifies the structure of the message. Figure 8 shows the same XML message as Figure 7, but it uses a more visual representation to highlight the message's structure.

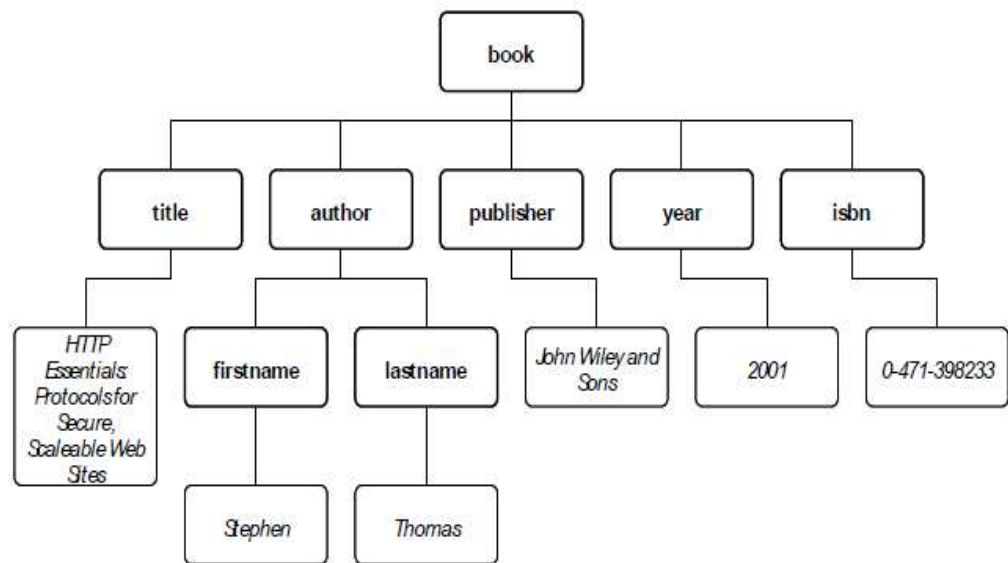


Figure 8. It is often convenient to show XML messages in a tree-like graph to highlight their structure.

The example of Figure 7 and Figure 8 is a relatively simple XML message with just a few elements. UPS Developer APIs rely on XML messages that are larger than this simple example, and it is difficult to show them in a pure graph like Figure 8 without losing legibility. Figure 9 shows an alternate way of depicting the tree-like structure of XML messages; it uses the same example as before.

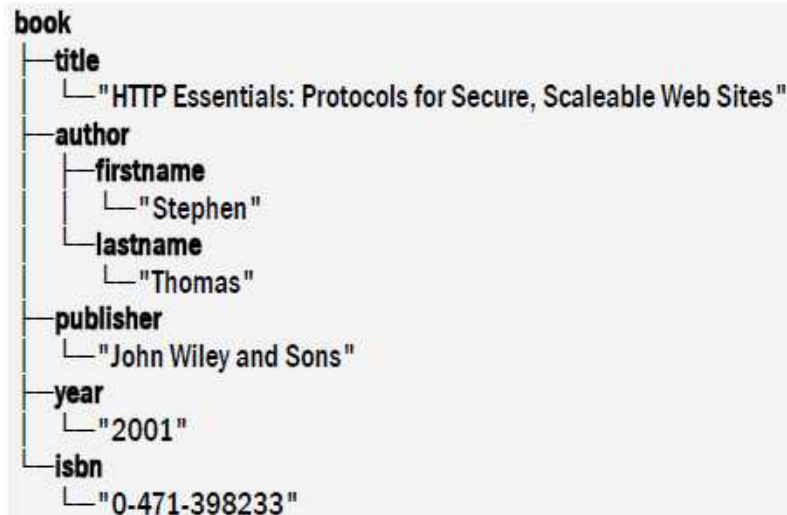


Figure 9. It is possible to show the tree-like structure of XML messages using text.

A great deal of information on XML is available on the Internet. A good starting point for further research is the World Wide Web Consortium's main page on XML at <http://www.w3.org/XML/>.

1.6.5 Working with XML in Application Programs

The popularity of Extensible Markup Language means that nearly all development environments include sophisticated support for creating and interpreting XML messages. This subsection provides a brief introduction to working with XML in Microsoft, Java, and Open Source environments. The examples it contains are only representative approaches; many other approaches are possible.

1.6.5.1 Microsoft Visual Studio

The Microsoft XML Core Services (MSXML) offers one way to work with XML documents in Visual Studio applications. Figure 10 shows one approach for creating an XML message in Visual Basic, and Figure 11 demonstrates how Visual Basic programs can interpret XML responses. To remain as simple as possible, the examples do not include any exception handling. Production software should, of course, provide full exception handling.

```
' Define a variable and initialize it to a new XML message
Dim dom
Set dom = New DOMDocument30

' Set properties of the variable
dom.async = False
dom.validateOnParse = False
dom.resolveExternals = False
dom.preserveWhiteSpace = True

' Identify the message as XML version 1.0
Set node = dom.createProcessingInstruction("xml", "version='1.0'")
dom.appendChild node
Set node = Nothing

' Create the root (book) element and add it to the message
Dim root
Set root = dom.createElement("book")
dom.appendChild root

' Create child elements and add them to the root
Dim node
Set node = dom.createElement("title")
node.text = "HTTP Essentials: ..."
root.appendChild node
Set node = Nothing

Set node = dom.createElement("author")
Dim child
Set child = dom.createElement("firstname")
child.text = "Stephen"
node.appendChild child
Set child = Nothing
Set child = dom.createElement("lastname")
child.text = "Thomas"
node.appendChild child
root.appendChild node

' And so on
```

Figure 10. Visual Basic can create XML messages through the DOMDocument object.

```

' Define a variable to hold the parsed message
Dim dom As New XmlDocument30
dom.async = False
dom.validateOnParse = False
dom.resolveExternals = False
dom.preserveWhiteSpace = True

' Try to parse the message
If dom.loadXML(messageText) = False Then
    ' The text did not contain valid XML
End If

' Get the title information from the message
Dim node As IXMLDOMNode
Set node = dom.selectSingleNode("/title")
If node Is Nothing Then
    ' The message did not contain title information
Else
    ' Do something with node.text
End If

```

Figure 11. Visual Basic can read the contents of XML messages after parsing them with the XmlDocument object.

1

Visual Basics also has built-in classes to interpret Base64-encoded values. The Convert.FromBase64String () function converts from a Base64-encoded string to an array of 8-bit unsigned integers.

1.6.5.2 Java Standard Edition and Enterprise Edition

The Java API for XML Processing (JAXP) is the preferred approach for working with XML messages in Java. Figure 12 shows how a BookClass object can be converted into an XML message, a process that JAXP calls *marshalling*. Similarly, Figure 13 shows the reverse process, where an XML message is converted into a BookClass object. This reverse process is *unmarshalling*. As above, exception handling code is omitted from these examples to make them as clear as possible.

```

import javax.xml.bind.JAXBContext;
import javax.xml.bind.Marshaller;
import java.io.StringWriter;

/* create the book object */
BookClass book = new BookClass("HTTP Essentials...",
    new AuthorClass("Stephen", "Thomas"),
    "John Wiley and Sons",
    "2001",
    "0-471-398233");

/* convert it to an XML string */
StringWriter writer = new StringWriter();
JAXBContext context = JAXBContext.newInstance(book.class);
Marshaller m = context.createMarshaller();
m.marshal(book, writer);

```

Figure 12. JAXP provides a convenient way for Java applications to create XML messages from Java objects.

```

import javax.xml.bind.JAXBContext;
import javax.xml.bind.Marshaller;
import java.io.StringReader;

/* xmlMessage contains the XML message */
StringReader reader = new StringReader(xmlMessage);
JAXBContext context = JAXBContext.newInstance(Book.class);
Unmarshaller u = context.createUnmarshaller();
BookClass book = (BookClass) u.unmarshal(reader);

/* access properties of the book */
System.out.println(book.getAuthor().getFirstname())

```

Figure 13. JAXP also gives Java applications a convenient way to retrieve elements from an XML message

1.6.5.3 Open Source

As of this writing, the most popular open source tool for working with XML is the xerces project. That project includes C++, Java, and Perl implementations of an XML library. The xerces distribution includes extensive sample applications. The code fragment in Figure 14 demonstrates creating an XML message in C++ with the xerces library. Figure 15 shows sample code for accessing an element within an XML message. In both cases exception handling is not shown in order to keep the examples as simple as possible. Production software should always contain full exception handling.

```

DOMImplementation* impl =
    DOMImplementationRegistry::getDOMImplementation(X("Core"));

DOMDocument* doc = impl->createDocument(0, X("book"), 0);
DOMElement* rootElem = doc->getDocumentElement();

DOMElement* titleElem = doc->createElement(X("title"));
rootElem->appendChild(titleElem);

DOMText* titleVal = doc->createTextNode(X("HTTP Essentials..."));
titleElem->appendChild(titleVal);

```

Figure 14. The Xerces C++ library includes classes that can create XML messages.

```

XMLCh* TAG_book;
XMLCh* TAG_title;
xercesc::XercesDOMParser *parser;

XMLPlatformUtils::Initialize();
TAG_book = XMLString::transcode("book");
TAG_title = XMLString::transcode("title");

parser = new XercesDOMParser;
parser->parse(inputText);
DOMDocument* xmlDoc = parser->getDocument();
DOMElement* elementRoot = xmlDoc->getDocumentElement();
DOMNodeList* children = elementRoot->getChildNodes();
const XMLSize_t nodeCount = children->getLength();
for (XMLSize_t cnt = 0; cnt < nodeCount; ++cnt) {
    DOMNode* currentNode = children->item(cnt);
    if( currentNode->getNodeType() &&
        currentNode->getNodeType() == DOMNode::ELEMENT_NODE ) {
        DOMElement* currentElement =
            dynamic_cast< xercesc::DOMElement* >( currentNode );
        if( XMLString::equals(currentElement->getTagName(),
                               TAG_title)) {
            /* do something with the title */
        }
    }
}

```

Figure 15. The Xerces C++ library also supports parsing XML messages to find individual elements.

1.7 Understanding XML Address Validation Services

With the Address Validation API, customers can validate the accuracy of their address at the regional level. The API can help reduce costly returns, provide better service to customers, and more accurately determine shipping costs.

To receive Address Validation information, an application sends a request to the Address Validation interface. The Address Validation request must contain one of the following input combinations:

- City, State, and Postal Code
- City
- Postal Code
- City and State
- City and Postal Code
- State and Postal Code

Once the request has been successfully submitted, the Address Validation API returns the following output data for each match (the closest match is returned first):

- Rank
- Quality
- City
- State
- Low-end postal code match
- High-end postal code match

When the Address Validation API finds close matches for a given input combination, a postal code range may be associated with each match. For example, suppose the input combination is **Atlanta, GA, 30300**. The Address Validation API would recognize that **30300** is an invalid postal code and would return up to 10 valid candidates that closely match the input combination. Some matches include a postal code range (for example, **30301-30371**); however, some matches only have one associated postal code, as illustrated in the table below. In this example, **30301** would be the low-end postal code match and **30371** would be the high-end postal code match.

Table 1. Returned Address Matches

Rank	Quality	City	State	Postal Range
1	0.926	Atlanta	GA	30301 - 30371
2	0.926	Atlanta	GA	30374 - 30381
3	0.926	Atlanta	GA	30383 - 30390
4	0.924	Atlanta	GA	30392 - 30392
5	0.924	Atlanta	GA	30394 - 30394
6	0.924	Atlanta	GA	30396 - 30396
7	0.924	Atlanta	GA	30398 - 30399
8	0.919	Atlanta	GA	31106 - 31107
9	0.919	Atlanta	GA	31150 - 31150
10	0.919	Atlanta	GA	39901 - 39901

1.7.1 Displaying the UPS Address Validation Notice

You must display the following notice, or such other language provided by UPS from time to time, in reasonable proximity to the Address Validation input and output information screens:

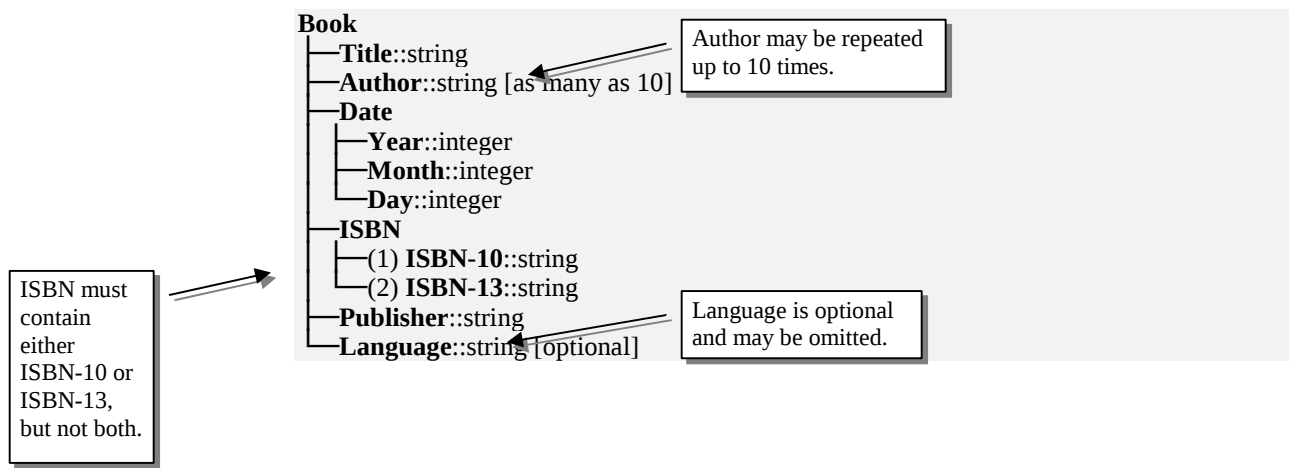
"NOTICE: UPS assumes no liability for the information provided by the address validation functionality. The address validation functionality does not support the identification or verification of occupants at an address.

1.8 The Address Validation XML Reference

This section documents the details of the XML messages, including the requests that clients send to UPS and the responses that UPS returns. The first subsection explains the notation that this section uses. It is followed by a section that describes how UPS verifies that client applications are authorized to use UPS API Services. The next subsections define the messages used by the API for its services. The final subsection provides a list of error codes.

Describing API Messages

As noted previously, all messages that UPS API Services send and receive consist of XML documents. This reference section defines the specific elements within those XML documents. Because XML documents follow a defined structure, this reference shows those elements using a compact, graphical notation. Here is an example of that notation, with some additional annotations to highlight important conventions:

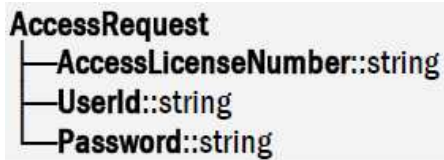


The figure indicates that a "Book" can contain six different child elements: Title, Author, Date, ISBN, Publisher, and Language. The Language element is marked "optional" so it is not required. All of the other five elements are required, however. Notice that the Author element can appear as many as ten times within the Book document. (The technical name for this property is *cardinality*.) Some of the child elements can themselves contain child elements. The Date element, for example, consists of a Year element, a Month element, and a Day element. Finally, notice that the children of the ISBN element (ISBN-10 and ISBN-13) have a number in parentheses preceding them. This number indicates a choice. In this example, the ISBN element must have either an ISBN-10 child element or an ISBN-13 child element, but it cannot have both. If an element has no child elements, the notation convention shows the type of content that the element can contain. In the example, Year, Month, and Day must contain integer numbers, while other primitive elements can contain arbitrary text strings.

Authenticating Client Applications

The Shipping API Services are only available for authorized UPS customers. To enforce this policy, UPS requires that every message that a client application sends include important authentication information. That information is contained in an `AccessRequest` XML document that must begin each message.

AccessRequest



Each `AccessRequest` contains three child elements: an `AccessLicenseNumber`, a `UserId`, and a `Password`. The following figure shows an example `AccessRequest` document.

```
<?xml version="1.0" ?>
<AccessRequest xml:lang='en-US'>
  <AccessLicenseNumber>
    YOURACCESSLICENSENUMBER
  </AccessLicenseNumber>
  <UserId>
    YOURUSERID
  </UserId>
  <Password>
    YOURPASSWORD
  </Password>
</AccessRequest>
```

An XML example of an Address Validation request follows:

```
<?xml version="1.0"?>
<AccessRequest>
  <AccessLicenseNumber>TEST262223144CAT</AccessLicenseNumber>
  <UserId>REG111111</UserId>
  <Password>REG111111</Password>
</AccessRequest>
<?xml version="1.0"?>
<AddressValidationRequest xml:lang="en-US">
  <Request>
    <TransactionReference>
      <CustomerContext>Maryam Dennis-Customer Data</CustomerContext>
      <XpciVersion>1.0001</XpciVersion>
    </TransactionReference>
    <RequestAction>AV</RequestAction>
  </Request>
  <Address>
    <City>MIAMI</City>
    <StateProvinceCode>FL</StateProvinceCode>
  </Address>
</AddressValidationRequest>
```

XML examples of successful Address Validation responses follow:

Example 1. Address Validation Response with a single match:

```
<?xml version="1.0"?>
<AddressValidationResponse>
  <Response>
    <TransactionReference>
      <XpciVersion>1.0001</XpciVersion>
    </TransactionReference>
    <ResponseStatusCode>1</ResponseStatusCode>
    <ResponseStatusDescription>Success</ResponseStatusDescription>
```

Address Validation Programming Information USPS Postal Addressing Standards

```
</Response>
  <AddressValidationResult>
    <Rank>1</Rank>
    <Quality>1.0</Quality>
    <Address>
      <City>TIMONIUM</City>
      <StateProvinceCode>MD</StateProvinceCode>
    </Address>
    <PostalCodeLowEnd>21093</PostalCodeLowEnd>
    <PostalCodeHighEnd>21094</PostalCodeHighEnd>
  </AddressValidationResult>
</AddressValidationResponse>
```

Example 2. Address Validation Response with multiple candidate matches:

```
<?xml version="1.0"?>
<AddressValidationResponse>
  <Response>
    <TransactionReference>
      <XpciVersion>1.0001</XpciVersion>
    </TransactionReference>
    <ResponseStatusCode>1</ResponseStatusCode>
    <ResponseStatusDescription>Success</ResponseStatusDescription>
  </Response>
  <AddressValidationResult>
    <Rank>1</Rank>
    <Quality>0.9975000023841858</Quality>
    <Address>
      <City>TIMONIUM</City>
```

```

        <StateProvinceCode>MD</StateProvinceCode>
    </Address>
    <PostalCodeLowEnd>21093</PostalCodeLowEnd>
    <PostalCodeHighEnd>21094</PostalCodeHighEnd>
</AddressValidationResult>
<AddressValidationResult>
    <Rank>2</Rank>
    <Quality>0.8299999833106995</Quality>
    <Address>
        <City>LUTHERVILLE TIMONIUM</City>
        <StateProvinceCode>MD</StateProvinceCode>
    </Address>
    <PostalCodeLowEnd>21093</PostalCodeLowEnd>
    <PostalCodeHighEnd>21094</PostalCodeHighEnd>
</AddressValidationResult>
<AddressValidationResult>
    <Rank>3</Rank>
    <Quality>0.8299999833106995</Quality>
    <Address>
        <City>LUTHERVILLE</City>
        <StateProvinceCode>MD</StateProvinceCode>
    </Address>

```

Address Validation Programming Information Address Validation Error Conditions

```

    <PostalCodeLowEnd>21093</PostalCodeLowEnd>
    <PostalCodeHighEnd>21094</PostalCodeHighEnd>
</AddressValidationResult>
</AddressValidationResponse>

```

1.8.1 AV Request

The Address Validation Request

The AddressValidationRequest message consists of two complete XML documents. The first document is an AccessRequest. It is immediately followed by an AddressValidation Request. When a client application is undergoing testing and integration, the document combination should be sent to the URL:

<https://wwwcie.ups.com/ups.app/xml/AV>

Once a client application is in production, the document combination should be sent to the URL:

<https://onlinetools.ups.com/ups.app/xml/AV>

The AddressValidationRequest document in request must confirm to following XML structure.

Name	XPath	Required	Max Allowed	Type	Length	Description
AddressValidationRequest	/AddressValidationRequest	Yes	1	Container	N/A	Container for AddressValidationRequest
Request	/AddressValidationRequest/Request	Yes	One	Container	N/A	Container for Request
TransactionReference	/AddressValidationRequest/Request/TransactionReference	No	One	Container	N/A	Transaction Reference container tag.
CustomerContext	/AddressValidationRequest/Request/TransactionReference/CustomerContext	No	One	String	1...512	Customer context can be any freeform, valid XML characters. The client uses CustomerContext to synchronize request/response pairs. The client establishes CustomerContext, which may contain customer-defined valid XML tags that are echoed in the response.
RequestAction	/AddressValidationRequest/Request/RequestAction	Yes	One	String	2	The action to be taken by the Address Validation tool. Must contain 'AV' to execute address validation.
Address	/AddressValidationRequest/Address	Yes	One	Container	N/A	Address container tag.
City	/AddressValidationRequest/Address/City	Cond	One	String	1...40	U.S. city to be validated. (A valid city/state/postal code combination must be included as input)

Name	XPath	Required	Max Allowed	Type	Length	Description
StateProvinceCode	/AddressValidationRequest/Address/StateProvinceCode	Cond	One	String	2	State to be validated. (A valid city/state/postal code combination must be included as input)
CountryCode	/AddressValidationRequest/Address/CountryCode	Yes	One	String	2	Two character Country Code. See Country Code table for values.
PostalCode	/AddressValidationRequest/Address/PostalCode	Cond	One	String	1...9	Postal code to be validated. (A valid city/state/postal code combination must be included as input)

1.8.2 AV Response

Address Validation Response

The Address Validation Response message contains a single XML document that conforms to the following XML structure.

Name	XPath	Required	Max Allowed	Type	Length	Description
AddressValidationResponse	/AddressValidationResponse	Yes	One	Container	N/A	Container for ValidationResponse
Response	/AddressValidationResponse/Response	Yes	One	Container	N/A	Container for Response
TransactionReference	/AddressValidationResponse/Response/TransactionReference	Yes	One	Container	N/A	Container for customer provided data and the XPCI Version.
CustomerContext	/AddressValidationResponse/Response/TransactionReference/CustomerContext	No	One	String	1...512	Customer context can be any free form, valid XML characters. The client uses CustomerContext to synchronize request/response pairs. The client establishes CustomerContext, which may contain customer-defined valid XML tags that are echoed in the response.
XpciVersion	/AddressValidationResponse/Response/TransactionReference/XpciVersion	No	One	String	1...6	Identifies the version of the message. The current XpciVersion is 1.0001.
ResponseStatusCode	/AddressValidationResponse/Response/ResponseStatusCode	Yes	One	String	1	Identifies the success or failure of the transaction. 1 = Successful 0 = Failed
ResponseStatusDescription	/AddressValidationResponse/Response/ResponseStatusDescription	No	One	String	1...10	Describes Response Status Code. Returns text of 'Success' or 'Failure'.

Name	XPath	Required	Max Allowed	Type	Length	Description
Error	/AddressValidationResponse/Response/Error	Yes	One	Container	N/A	If an error is encountered during the interchange, the Response contains an error. If the error is present, then the ErrorSeverity and ErrorCode are required.
ErrorSeverity	/AddressValidationResponse/Response/Error/ErrorSeverity	Yes	One	String	1...10	Describes the severity of the error. \n\nTransientError â “Customer's data has not been processed due to system unavailability. The customer has to wait and try again. HardError - The error was encountered processing the customer's data and that the data needs correction. Warning - The customer's data was successfully processed; however, there were warnings encountered during processing.
ErrorCode	/AddressValidationResponse/Response/Error/ErrorCode	Yes	One	String	2	A numeric value that describes the error. Each tool defines a range of error codes.
ErrorDescription	/AddressValidationResponse/Response/Error/ErrorDescription	No	One	String	1...150	Describes the error code.
MinimumRetrySeconds	/AddressValidationResponse/Response/Error/MinimumRetrySeconds	No	One	String	1...10	Number of seconds to wait until retry. This field is populated on special conditions of the Transient Error only, as defined by the service. A number between 1 and 86400 (24 hours)
ErrorLocation	/AddressValidationResponse/Response/Error/ErrorLocation	Yes	One	Container	N/A	Identifies the element in error.

Name	XPath	Required	Max Allowed	Type	Length	Description
ErrorLocationElementName	/AddressValidationResponse/Response/Error/ErrorLocation/ErrorLocationElementName	No	One	String	1...30	The XPATH name of the element in error. This is a valid XPATH pointing to an element in the request document.
ErrorLocationAttributeName	/AddressValidationResponse/Response/Error/ErrorLocation/ErrorLocationAttributeName	No	One	String	1...30	The name of the attribute in error. This is the name of the attribute contained by the ErrorLocation Element.
ErrorDigest	/AddressValidationResponse/Response/Error/ErrorDigest	No	Many	String	unbounded	The contents of the element in error.
AddressValidationResult	/AddressValidationResponse/AddressValidationResult	Yes	Many	Container	N/A	Address Validation Result container tag.
Rank	/AddressValidationResponse/AddressValidationResult/Rank	Yes	One	String	1...10	The rank of each range result.
Quality	/AddressValidationResponse/AddressValidationResult/Quality	Yes	One	String	2	The quality factor, which describes the accuracy of the result compared to the request. 1.0 = Exact match. 95-.99 = Very close match. 90-.94 = Close match. 70-.89 = Possible match. 00-.69 = Poor match

Name	XPath	Required	Max Allowed	Type	Length	Description
Address	/AddressValidationResponse/AddressValidationResult/Address	Yes	One	Container	N/A	Container for address
City	/AddressValidationResponse/AddressValidationResult/Address/City	Yes	One	String	1...30	The city for this address match.
StateProvinceCode	/AddressValidationResponse/AddressValidationResult/Address/StateProvinceCode	Yes	One	String	2	The state for this address match.
PostalCodeLowEnd	/AddressValidationResponse/AddressValidationResult/PostalCodeLowEnd	Yes	One	String	1...10	When the Address Validation tool finds matches for a given input combination, a postal code range may be associated with each match. This is the low end of the range.
PostalCodeHighEnd	/AddressValidationResponse/AddressValidationResult/PostalCodeHighEnd	Yes	One	String	1...10	When the Address Validation tool finds matches for a given input combination, a postal code range may be associated with each match. This is the high end of the range.

1.9 Customer Integration Environment

The Customer Integration Environment allows customers to test their application prior to launch. This environment is intended for integration testing of customer applications with the UPS servers. No stress testing should ever be performed by customers against any UPS systems.

Once your application has been thoroughly tested, you should redirect the application to the UPS Production Environment.

Please note that while the Customer Integration Environment maintains system availability 24 hours, 7 days each week, there are occasional system down times to allow for server maintenance.

1.9.1 Address Validation

Test your Address Validation application with valid and invalid address elements. It is recommended that you use addresses that are familiar to you, for example, your home or business address. This will ensure that your application has the ability to process success and error responses correctly.

For integration testing, you should direct your Address Validation software to

<https://wwwcie.ups.com/ups.app/xml/AV>.

Once you have completed your testing, you should direct your software to the production URL below:

<https://onlinetools.ups.com/ups.app/xml/AV>.

1.9.2 System Availability

The Customer Integration Environment is available 24 hours a day, 7 days a week.

1.9.3 Server Availability Check

All of the UPS services work using HTTPS POST. Using the same URL as you point your application to, perform an HTTP GET. If the server is available, it will reply with the service name, remote user, server port, server name and servlet path. To see this in action, type the following URL in your web browser:

<https://onlinetools.ups.com/ups.app/xml/AV>

You should see the following in the browser window:

Service Name: AV

Remote User: null

Server Port: 443

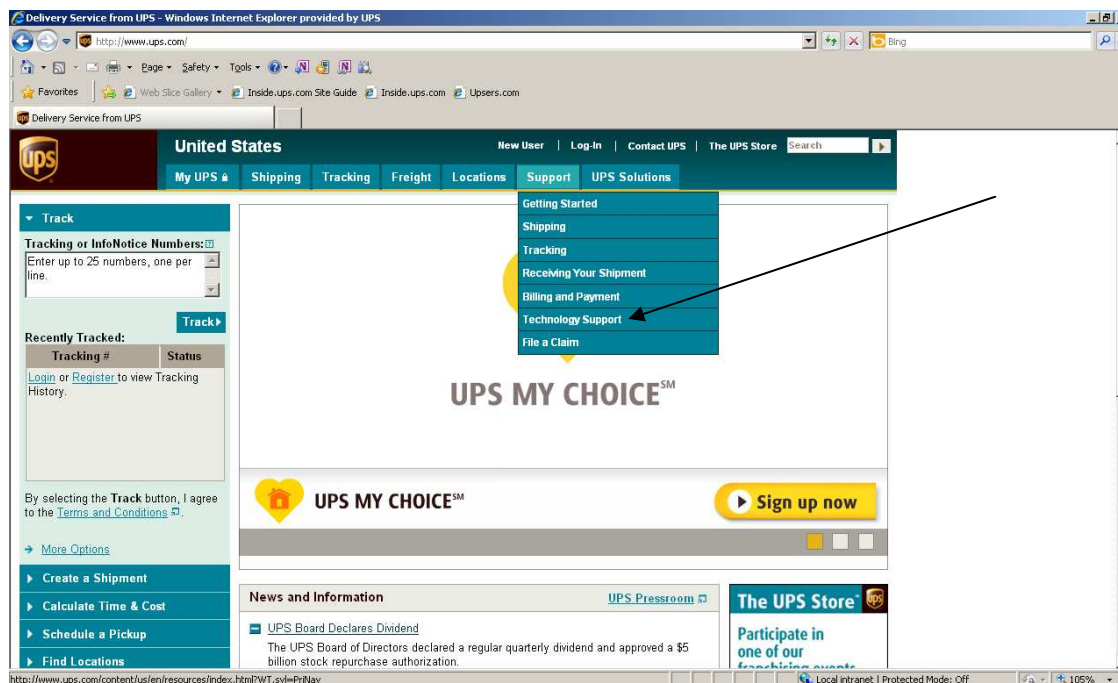
Server Name: onlinetools.ups.com

Servlet Path: /AV

Appendix A - Accessing Tech Support (US Site Example)

Go to ups.com and log in to myUPS with your ID and PW.

Hover over the “Support” tab, and select “Technology Support” from the drop-down.



From the Technology Support page select the Developer Resource Center link from the left navigation.

UPS: Technology Support - Windows Internet Explorer provided by UPS

http://www.ups.com/content/us/en/resources/techsupport/index.html?WT.svl=PNRO_L1

UPS: Technology Support

United States

New User | Log-In | Contact UPS | The UPS Store

My UPS | Shipping | Tracking | Freight | Locations | Support | UPS Solutions

Support

- Getting Started
- Shipping
- Tracking
- Receiving Your Shipment
- Billing and Payment
- Technology Support
- Developer Resource Center
- UPS WorldShip Support
- Host Customer Info Center

Search Support

Enter a keyword:

Search

UPS MY CHOICE™

TIRED OF GETTING THIS?

Sign up now

Technology Support

How To...

Get helpful information on how to use ups.com.

- Reset My Password
- Retrieve Username
- Register for UPS.com
- Use Your UPS Thermal Printer
- Get UPS Billing Data
- Download the UPS Billing Analysis Tool
- Access the Developer Resource Center
- View My Shipping History
- Request a Delivery Intercept
- Import/Export a File
- Process a Return Shipment
- Void a Shipment
- Process a Shipment
- Reprint a Label
- Reprint a Report
- Run End-Of-Day in WorldShip
- Add Shipper Number

Tools and Resources

Know what tool you need? Get started now.

- Calculate Time and Cost
- WorldShip® Support
- UPS Developer Kit
- UPS WorldShip Help Files
- Online Shipping Demo
- Host Customer Info Center

UPS Ready Program

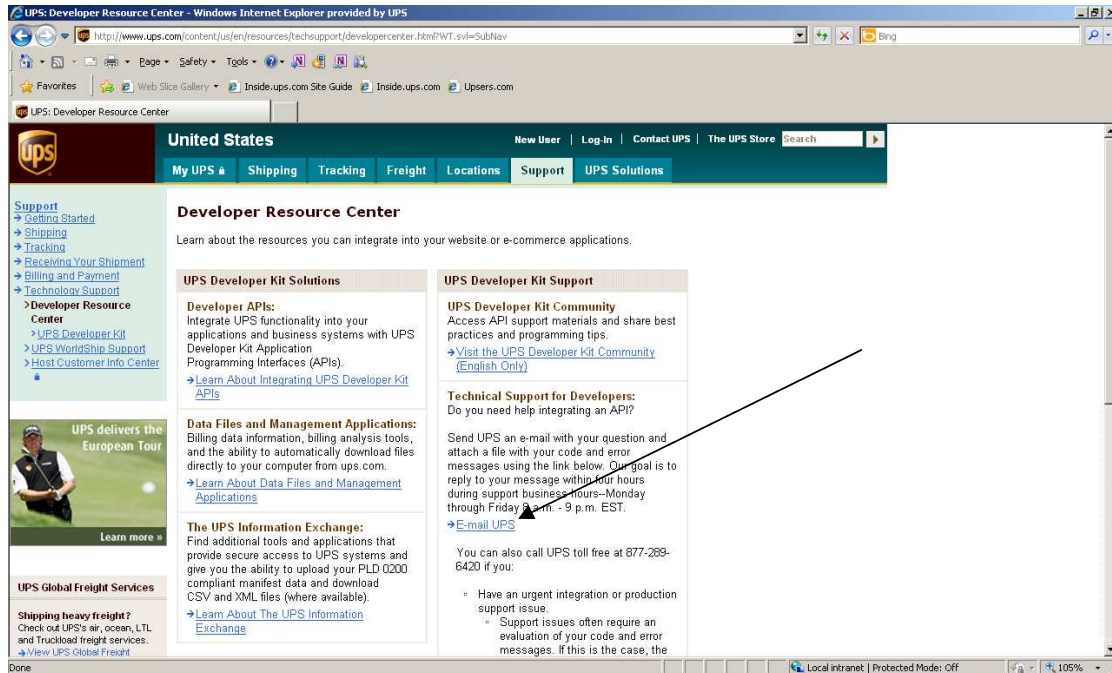
The UPS Ready® Program offers access to UPS through technology solutions. Choose below to learn more.

- UPS Ready Program

Find it on ups.com

Still looking for the support information you need? Enter a phrase, keywords, or a

From the Developer Resource Center page select the “Email UPS” link in right hand column.



Complete Name, email address, Support Category must be “Technical Support”, and Support Topic must be “Developer Resource”. Select the “Next” button.

Scroll down the page and complete the remainder of the email form including “Your Telephone”, “Stage of Development”, “Developer Resource” which is which API you are integrating, “Attach File” where you attach your XML Request/Response Pair, and any pertinent description of the issues in the “What is your question or comment?” field. Select “Send Email” button.

UPS: E-mail UPS - Windows Internet Explorer provided by UPS

https://www.ups.com/upsemal/includeForm?loc=en_US

United Parcel Service Inc [US]

UPS: E-mail UPS

Please enter the following information to help us provide a better answer to your e-mail question. Select **Send E-mail** to continue, and UPS will respond within one business day. Required fields are indicated with *.

Your Telephone:

Stage of Development:

Developer Resource:

Version Number:

Attach File:

If you need to attach multiple files, please combine them into a single archive file (e.g. a zip file) and attach the archive file. The file must not exceed 2.9 MB.

What is your question or comment?

Type description of problem here.

(Maximum 1000 characters)

The screen below will appear. Please see “Technical Support (US)” within this document for tech support hours of operation and response times.

UPS: E-mail UPS - Windows Internet Explorer provided by UPS

https://f1efxer.ups.com/upsemal/onlineTools/loc=en_US

United States

Welcome, Daniel Franz | Logout | Contact UPS | The UPS Store

My UPS | Shipping | Tracking | Freight | Locations | Support | UPS Solutions

E-mail UPS

Your e-mail has been sent to UPS.
A customer service representative will respond to you within one business day.

UPS MY CHOICE™

TIRED OF GETTING THIS?

UPS Global Freight Services

Shipping heavy freight?
Check out UPS's air, ocean, LTL and Truckload freight services.
[View UPS Global Freight Services Demo](#)

Subscribe to UPS E-mail: [Modify E-mail Preferences](#) [View Examples](#)

[Site Feedback](#)

Appendix B – Address Validation Error Codes

To discover errors, check the ResponseStatusCode element. A “1” normally indicates a successful response, whereas a “0” indicates an error, either Transient or Hard. When an error occurs there will also be an error code, and an error description.

- Success – Successful responses may or may not include **Warnings**.
 - (without warnings) Request is processed as anticipated by the client.
 - (with warnings) *Warning* messages indicate that UPS was able to process the request; however (potentially) unanticipated results have also occurred. The warning contains information in the response that should be passed to the end user.
- Errors – will return two different levels of severity.
 - *Transient* errors are temporary errors, due to temporary high server loads or scheduled maintenance, for example. The application may re-issue the request at a later time.
 - *Hard* errors indicate that an error existed in the request that UPS could not resolve. These errors are critical and prevent requests from processing.

Applications should not re-issue requests with Hard errors without first correcting the error. The following table lists the errors that UPS may return in response to a request.

Error Code	Severity	Description	Condition
101111	Transient	General Address Validation Failure	Address Validation system temporarily unavailable
101112	Hard	No Address Candidate Found	Address candidate not found
10001	Hard	The XML document is not well formed	
10002	Hard	The XML document is well formed but the document is not valid	
260050	Hard	Invalid RequestAction	The request action is invalid.
264001	Transient	AV Service is not available	Adapter parsing error, business process calling error, backend service is unavailable, etc.
264002	Hard	Country Code is invalid or missing.	
264003	Hard	The Maximum allowable Candidate List size has been exceeded within the User Request.	The maximum candidate list size requested from the user has been exceeded.
264004	Hard	The maximum validation query time has been exceeded due to poor address data.	Request has timed out. Usually due to insufficient or poor address data from client.
264005	Hard	Address classification is not valid for a regional request.	The customer submits a request for address classification with a regional address format.

Error Code	Severity	Description	Condition
264006	Hard	Invalid candidate list size.	The maximum candidate list size given by the customer is not a numeric value between 0 and 2147483647.
264007	Hard	Address classification is not allowed for the country requested	The request contains address classification with a country code that is not supported for classification
264008	Hard	Country code and address format combination is not allowed.	Country code and address format combination is not allowed.
264027	Hard	Additional address fields are needed to perform the requested operation.	The Country Code is valid but the other fields are blank. Additional fields need to be provided in order to validate or classify the address.

Appendix C - Country Codes

UPS country code abbreviations generally follow the recommendations of the International Standards Organization, which publishes a list of currency abbreviations in ISO Standard 3166. The following table lists the ISO country codes that ISO had defined when this document was published. The latest information is available from the ISO web site.

Please note that not all UPS services are available in every country. For more information on UPS services, refer to the latest *UPS Rate and Service Guide* available at <http://www.ups.com>.

Country Code	Country Name	Classification	Validation
US	United States	X	X
PR	Puerto Rico		X
CA	Canada	X	
AL	Albania		
DZ	Algeria		
AS	American Samoa		
AD	Andorra		
AO	Angola		
AI	Anguilla		
AG	Antigua and Barbuda		
AR	Argentina		
AM	Armenia		
AW	Aruba		
AU	Australia		
AT	Austria		
AZ	Azerbaijan		
AP	Azores		
BS	Bahamas		
BH	Bahrain		
BD	Bangladesh		
BB	Barbados		
BY	Belarus		
BE	Belgium		
BZ	Belize		
BJ	Benin		
BM	Bermuda		
BT	Bhutan		
BO	Bolivia		
BL	Bonaire		
BA	Bosnia and Herzegovina		
BW	Botswana		
BV	Bouvet Island		
BR	Brazil		
IO	British Indian Ocean Territory		

Country Code	Country Name	Classification	Validation
BN	Brunei Darussalam		
BG	Bulgaria		
BF	Burkina Faso		
BI	Burundi		
KH	Cambodia		
CM	Cameroon		
CD	Channel Islands		
IC	Canary Islands		
CV	Cape Verde		
KY	Cayman Islands		
CF	Central African Republic		
TD	Chad		
CL	Chile		
CN	China		
CX	Christmas Island		
CC	Cocos (Keeling) Islands		
CO	Colombia		
KM	Comoros		
ZP	Congo (Democratic Republic of Congo)		
CK	Cook Islands		
CR	Costa Rica		
CI	Cote D' Ivoire (Ivory Coast)		
HR	Croatia (Hrvatska)		
CB	Curacao		
CY	Cyprus		
CZ	Czech Republic		
DK	Denmark		
DJ	Djibouti		
DM	Dominica		
DO	Dominican Republic		
TP	East Timor		
EC	Ecuador		
EG	Egypt		
SV	El Salvador		
EN	England		
GQ	Equatorial Guinea		
ER	Eritrea		
EE	Estonia		
ET	Ethiopia		
FO	Faeroe Islands		
FK	Falkland Islands (Malvinas)		
FJ	Fiji		
FI	Finland		

Country Code	Country Name	Classification	Validation
FR	France		
GF	French Guiana		
PF	French Polynesia		
TF	French Southern Territories		
GA	Gabon		
GM	Gambia		
GE	Georgia		
DE	Germany		
GH	Ghana		
GI	Gibraltar		
GB	Great Britain (UK)		
GR	Greece		
GL	Greenland		
GD	Grenada		
GP	Guadeloupe		
GU	Guam		
GT	Guatemala		
GN	Guinea		
GW	Guinea-Bissau		
GY	Guyana		
HT	Haiti		
HM	Heard Island and McDonald Islands		
HN	Honduras		
HK	Hong Kong		
HU	Hungary		
IS	Iceland		
IN	India		
ID	Indonesia		
IE	Ireland		
IL	Israel		
IT	Italy		
JM	Jamaica		
JP	Japan		
JO	Jordan		
KZ	Kazakhstan		
KE	Kenya		
KI	Kiribati		
KO	Kosrae		
KW	Kuwait		
KG	Kyrgyzstan		
LA	Laos		
LV	Latvia		
LB	Lebanon		

Country Code	Country Name	Classification	Validation
LS	Lesotho		
LR	Liberia		
LY	Libya		
LI	Liechtenstein		
LT	Lithuania		
LU	Luxembourg		
MO	Macau		
MK	Macedonia		
MG	Madagascar		
ME	Madeira		
MW	Malawi		
MY	Malaysia		
MV	Maldives		
ML	Mali		
MT	Malta		
MH	Marshall Islands		
MQ	Martinique		
MR	Mauritania		
MU	Mauritius		
YT	Mayotte		
MX	Mexico		
FM	Micronesia		
MD	Moldova		
MC	Monaco		
MN	Mongolia		
MS	Montserrat		
MA	Morocco		
MZ	Mozambique		
MM	Myanmar		
NA	Namibia		
NR	Nauru		
NP	Nepal		
NL	Netherlands		
AN	Netherlands Antilles		
NT	Neutral Zone		
NC	New Caledonia		
NZ	New Zealand (Aotearoa)		
NI	Nicaragua		
NE	Niger		
NG	Nigeria		
NU	Niue		
NF	Norfolk Island		
KP	North Korea		
NB	Northern Ireland		

Country Code	Country Name	Classification	Validation
MP	Northern Mariana Islands		
NO	Norway		
OM	Oman		
PK	Pakistan		
PW	Palau		
PA	Panama		
PG	Papua New Guinea		
PY	Paraguay		
PE	Peru		
PH	Philippines		
PN	Pitcairn		
PL	Poland		
PO	Ponape		
PT	Portugal		
SA	Saudi Arabia		
QA	Qatar		
RE	Reunion		
RO	Romania		
RT	Rota		
RU	Russian Federation		
RW	Rwanda		
SS	Saba		
KN	Saint Kitts and Nevis		
LC	Saint Lucia		
VC	Saint Vincent and the Grenadines		
SP	Saipan		
WS	Samoa		
SM	San Marino		
ST	Sao Tome and Principe		
SF	Scotland		
SN	Senegal		
CS	Serbia and Montenegro		
SC	Seychelles		
SL	Sierra Leone		
SG	Singapore		
SK	Slovak Republic		
SI	Slovenia		
SB	Solomon Islands		
SO	Somalia		
ZA	South Africa		
GS	South Georgia and South Sandwich Islands.		
KR	South Korea		

Country Code	Country Name	Classification	Validation
ES	Spain		
LK	Sri Lanka		
NT	St. Barthelemy		
SW	St. Christopher		
VI	St. Croix		
EU	St. Eustatius		
SH	St. Helena		
UV	St. John		
KN	St. Kitts and Nevis		
LC	St. Lucia		
MB	St. Maarten		
TB	St. Martin		
PM	St. Pierre and Miquelon		
VL	St. Thomas		
VC	St. Vincent/Grenadine		
SD	Sudan		
SR	Suriname		
SJ	Svalbard and Jan Mayen Islands		
SZ	Swaziland		
SE	Sweden		
CH	Switzerland		
SY	Syria		
TA	Tahiti		
TW	Taiwan		
TJ	Tajikistan		
TZ	Tanzania		
TH	Thailand		
TI	Tinian		
TG	Togo		
TK	Tokelau		
TO	Tonga		
TL	Tortola		
TT	Trinidad and Tobago		
TU	Truk		
TN	Tunisia		
TR	Turkey		
TM	Turkmenistan		
TC	Turks and Caicos Islands		
TV	Tuvalu		
UG	Uganda		
UA	Ukraine		
UI	Union Island		
AE	United Arab Emirates		

Country Code	Country Name	Classification	Validation
UY	Uruguay		
UM	US Minor Outlying Islands		
SU	USSR (former)		
UZ	Uzbekistan		
VU	Vanuatu		
VA	Vatican CityState (Holy See)		
VE	Venezuela		
VN	Vietnam		
VR	Virgin Gorda		
VG	Virgin Islands (British)		
VI	Virgin Islands (U.S.)		
WL	Wales		
WF	Wallis and Futuna Islands		
WS	Western Samoa		
YA	Yap		
YE	Yemen		
ZR	Zaire		
ZM	Zambia		
ZW	Zimbabwe		

Appendix D - Frequently Asked Questions: Address Validation

API	Category	Question	Answer
Address Validation	Address Validation	How often is AV data updated?	Currently the UPS Address Validation API's database is updated monthly with new address information from USPS. The database updates should synchronize the information with the USPS. Addresses not currently available through UPS will be added once they are added to USPS.
Address Validation	General	What countries' addresses can be validated by the Address Validation API?	The Address Validation API can only be utilized to validate addresses in the regional level (city, state and zip code) within the US.
Address Validation	General	How much time will it take to program/implement the Address Validation API or the Address Validation - Street Level API?	The programming/implementation of the Address Validation or the Address Validation - Street Level APIs may vary and is strictly dependent on the skill level of the developer. An implementation of Address Validation Street Level may take as little as a week for a very skilled developer to as long as months for a less skilled developer.
Address Validation	Address Validation frequency of updates	How frequently are the databases updated for Address Validation Street Level and Address Validation APIs?	Street Level Address Validation 1. Classification = weekly on Sunday input from operations. 2. Validation = monthly day varies. Address Validation 1. No classification 2. Validation updated every six months.
Address Validation	Batch Upload	Does either the Address Validation or Address Validation Street Level APIs offer the ability to batch upload?	No. Neither API provides batch upload. Only individual requests.
Address Validation	Suite/Apt #	Do either of the AV or AVSL APIs provide a candidate list for addresses that have suite/apt information?	The Tool does not return candidate lists for suite or apartment number ranges when they are not provided by the customer.

API	Category	Question	Answer
All	General - security	Does UPS support chained or unchained digital certificates?	Currently, the Ship API uses an unchained cert which will be migrated to chained Sept '09. In the case of the Ship API, INET is responsible for those urls and corresponding Digital Certs. They are being renewed at the end of this month as unchained. They will migrate to chained in Sept '09.
All	API availability	Within what countries are the Developer APIs available?	The Developer APIs are available in the countries listed at ups.com by API and by country under the UPS Developer Kit - Developer APIs. Click on any particular API to go to the API page and scroll to the bottom of the page to get the country listing for that particular API.
All	Mail Innovations	Is Mail Innovations available within any of the Developer APIs?	There is no UPS API available for Mail Innovations services at this time nor did we see one at http://www.upsmailinnovations.com . Additionally we are not aware of any plans to support the services through the UPS Developer APIs. The customer may however contact a Mail Innovations representative by clicking the Support tab of the site for any additional questions or concerns.
All	UPS Logos High Resolution	How do I get access to higher resolution logos than what are available with downloads for the UPS Developer Kit - Developer APIs?	Customers are not routinely provided these UPS shield graphics without having a design/layout submitted to UPS Brand Management for approval. The customer needs to visit: https://www.upsbrandexchange.com/brandHome.awsp This site will take them through the process for downloading a limited set of sample images for layout, and how to secure an approval for customer use, as well as the high-resolution graphics.
All	Technical Support email form	Is XPCI a required field within the email support form?	No.

API	Category	Question	Answer
All	Technical Support email form	From where is the XPCI version number obtained and what does it mean?	XPCI stands for XML Package Carrier Interface (XPCI) and defines a vocabulary and structure for describing packages, shipments, and the activity details for package carriers and their customers. XPCI is a set of DTDs that defines the terminology, transaction enveloping, and XML message definitions. For a client to be XPCI-compliant, the client must generate a well-formed XML message that validates against the XPCI DTDs. Several DTDs, organized into three categories, define XPCI: Vocabulary — This DTD defines the basic business vocabulary of XPCI. All tags used in a message are defined in this DTD. Interchange — This DTD defines the transaction-enveloping scheme. Every message includes transaction information. Message — Each message has an associated DTD that defines the vocabulary of the message. The version and date would have been related to versioning however the APIs were not versioned so they currently do not carry significance. They remain as part of the APIs so that in the event they are versioned, we have these elements “just in case”.
All	Technical Support	How do I get technical support for the APIs at ups.com?	Go to the Developer Resource Center and select email support under the UPS Developer Kit Support Column.
All	Characters	Can Japanese Kanji character be recognized by UPS Developer APIs?	No.
All	Basic	Do any of the Developer APIs support Basic service?	No. Basic is not supported within the Rating or Shipping APIs but is supported within the Tracking API available within the UPS Developer Kit - Developer APIs.

API	Category	Question	Answer
All	System Down-Times	Are there any designated systems down times for the Developer APIs?	Yes. The overall reserved downtime for the CGI servers is Saturday 10:00 PM ET through Sunday 12:00 PM ET. However, often the window is shortened to two 15 minute intervals with one starting at 11 PM and the other occurring sometime between 1 and 3 AM ET Sunday morning for most weekends. The back end goes through numerous updates typically beginning at 11:00PM Saturday through 4:00AM Sunday. Typically traffic is handled in such a way that there is very little impact to customers, and any impact which does occur does so in the small 15 minute intervals mentioned previously. Having said this as this entire time is reserved for maintenance we inform customers of the possibility of experiencing issues throughout this time period so that if there are any issues which occur during maintenance we have a time window to troubleshoot and perform measures to resolve. On Sunday, the maintenance is really relegated to just ABR and freight.
All	ASMX	Are the Web Services versions of the APIs ASMX based?	No. All Web Services are XML based. This is described in the section 'UPS OnLine Tools Technologies' of every developer's guide
All	Web Services - Empty folders within the documentation zip file.	The ship_dev_guide and Ship_Reference_guide folders have some sub folders that look like they should contain some code examples / samples but they are all empty? XML_Samples Visual_Basic Code_samples All empty?	Unfortunately code samples are not provided with the Shipping API - Web Services version. The reason being is that a WSDL is included which provides all of the necessary information needed to successfully implement the API. These folders are typically utilized in the XML version of the APIs as there is no WSDL present. If the customer wishes to view the samples contained in the Shipping API they can download the documentation by logging into UPS.com, navigating to the UPS Developer Kit, and then clicking on the Shipping API link.

API	Category	Question	Answer
All	Pointing to the wrong URL for API	I keep getting, "XML document is well formed but the document is not valid." error message. What am I doing wrong?	The "XML document is well formed but the document is not valid" error message is generally returned when an element in the XML request does not adhere to the formatting defined within the Xpath section of that API's developer guide. When the API returns this error it indicates the field which is not valid in the ErrorLocationElementName element in the XML response. When we test the XML provided by you earlier in this email chain we are able to receive a successful response. This would indicate that you may be posting to an incorrect URL. The error message returned from the API should have contained a line similar to the following: <pre><ErrorLocationElementName>XPATH TO FIRST ELEMENT WHERE XML DOESN'T MATCH EXPECTED FORMAT</ErrorLocationElementName></pre> As previously stated, this element is included in the XML error response to point you to the element of the posted request which is not valid for the Tool. When further clarification is needed you can look up the element in question in the Xpath section of the Developer Guide for the particular API. If this element lists another API's request such as "TrackRequest" it would indicate that you are posting to that API's URL and need to adjust the URL you are sending your XML to.
All	Phone Support	Is phone support provided for the UPS Developer Kit - Developer APIs? If so, what is the number and what are the hours of operation?	Yes. Phone support is provided at 1st Level only and for basic API questions. This includes integration questions and production questions. However, customers' questions that cannot be answered verbally will be directed to the email support form at ups.com to escalate to 3rd level via email. Phone Support Hours: M-F 730am- 9pm EST Sa-Su 9am - 6pm EST 1800-247-9035.

API	Category	Question	Answer
All	Examples of API Implementations	Are there any examples of implementations that we can review to understand how best to utilize the APIs?	We do not share customer implementations of our tools amongst customers. On occasion we do post case studies on ups.com and articles in customer-facing newsletters, but that is only after gaining permission from the customer and working with Legal, Customer Communications, etc. Please understand that the XML tool is only data, which is transparent to the end user. How the developer implements the tool and presents results back to the end user can vary from web site to web site. These web sites may not highlight the full functionality of the tool. We need to be able to describe the value proposition of the tool without depending on another customer's usage.
All	Code languages supported	Do the APIs support PHP or Perl with code sample within the Developer Guides or the developer kit zip files?	Yes. We currently support PHP or Perl with sample code.