Address Validation Street Level XML Developers Guide

December 30, 2013

Notice

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

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 6
1.3 Business Processes and Rules 7
1.4 Required Steps for Integrating 7
1.5 Planning Your Applications for XML 8
1.5.1 UPS Developer API Applications 8
1.5.2 Licensing the UPS Developer APIs 9
1.5.2.1 Branding Requirements 10
1.5.3 Developing Client Applications for Developer APIs 10
1.5.3.1 Microsoft Visual Studio 10
1.5.3.2 Java Standard Edition and Enterprise Edition 10
1.5.3.3 Open Source 10
1.5.4 Getting Technical Support 11
1.5.5 Testing and Deploying Applications 14
1.5.6 Keeping Up-to-Date 14
1.6 UPS Developer API Technologies for XML 14
1.6.1 Hypertext Transfer Protocol and Secure Sockets Layer 3 14
1.6.2 Working with HTTP and SSL in Application Programs 16
1.6.2.1 Microsoft Visual Studio 17
1.6.2.2 Java Standard Edition and Enterprise Edition 18
1.6.2.3 Open Source 18
1.6.2.4 SSL Certificate Changes and Renewals 19
1.6.3 Extensible Markup Language (XML) 20
1.6.4 Working with XML in Application Programs 23
1.6.4.1 Microsoft Visual Studio 23
1.6.4.2 Java Standard Edition and Enterprise Edition 25
1.6.4.3 Open Source 26
1.7 Understanding XML Address Validation Street Level Services 27
1.7.1 Address Validation 27
1.7.2 Address Classification 29
1.8 The Address Validation Street Level XML Reference 30
1.8.1 Describing UPS API Services Messages 30
1.8.2 Authenticating Client Applications 31
1.8.3 XAV Request 33
1.8.4 XAV Response 37
1.9 Customer Integration Environment 42
1.9.1 Street Level Address Validation 42
1.9.2 System Availability 42
1.9.3 Server Availability Check 42

Appendix A - Accessing Tech Support (US Site Example) 43
Appendix B – Address Validation Street Level Error Codes 47
XML Errors 48
Common Architecture Errors 48
Common License and Access Control Errors 48
Appendix C - Country Codes 49
Appendix D - Country Classification and Validation Matrix 56
Appendix E - Frequently Asked Questions: Address Validation Street Level 57

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 2014	No changes
Jul 2013	No changes

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 Street Level API for only packages manifested, tendered, and delivered by UPS.

- Any customers/developers abusing the Address Validation Street Level API or data mining the API will have their access revoked.
- The Address Validation Street Level API provides residential/commercial classification based on the information provided by the UPS driver network.
- UPS initially provides testing privileges for Address Validation Street Level API. Production Access to the API needs to be requested from the Developer Resource Center on UPS.com using the Access Key already provided.
- To obtain testing and/or production access to the Address Validation Street Level 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.
- Authentication - 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. Add your functionality specific to your application and/or web site to the skeleton code.
4. Test your application and/or web site using the designated UPS staging environment.
5. If you are a UPS Ready developer, review your application with UPS.
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.

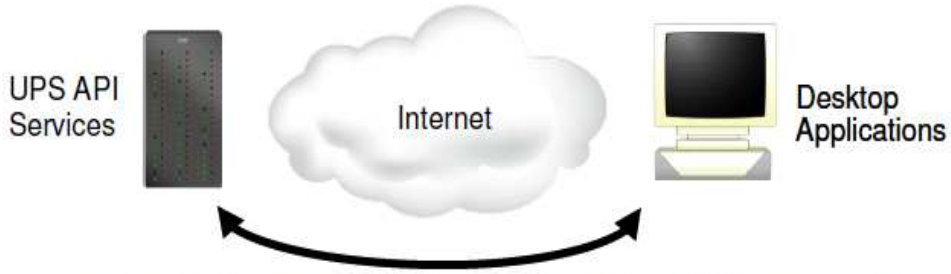


Figure 1: Dedicated applications that users run on their desktops can access UPS Developer APIs.

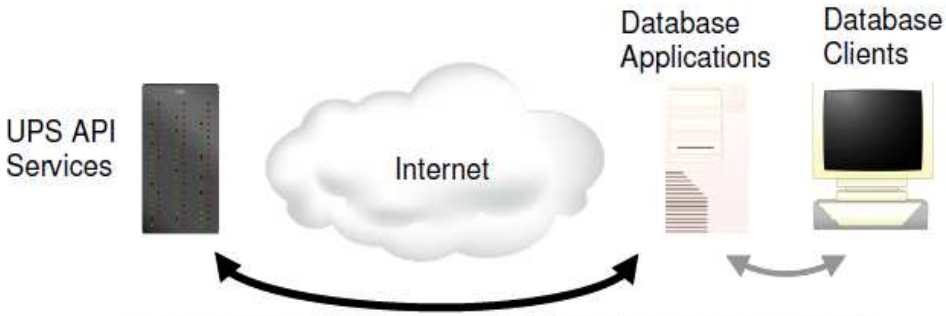


Figure 2: Database applications can access UPS Developer APIs and return information to their clients

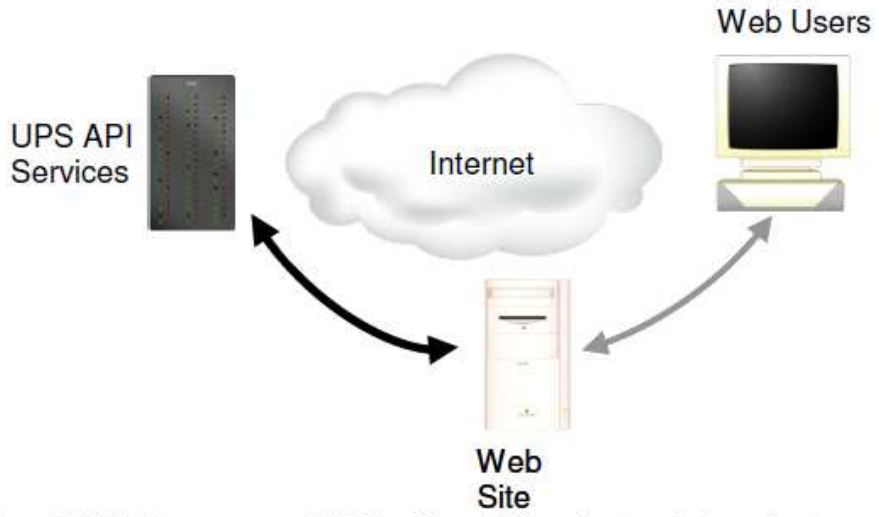


Figure 3: Web sites can access UPS Developer APIs and return information to users' web browsers.

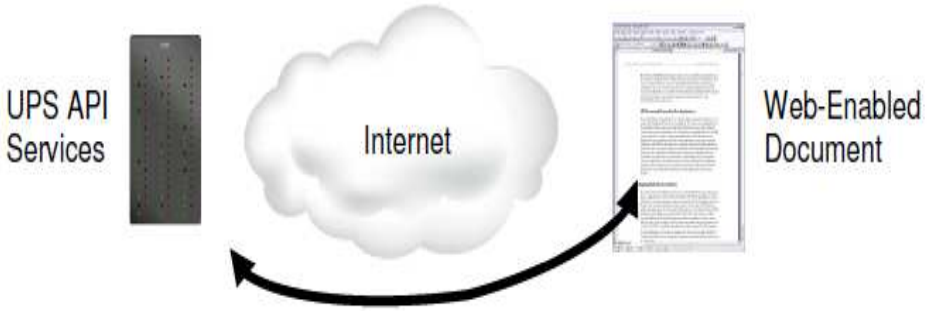


Figure 4: Non-traditional applications such as Microsoft Office or Adobe Acrobat documents can use UPS Developer APIs to automatically update their content.

The UPS Developer APIs can be implemented withal 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. Look for the icons in the left margin, which identify information relevant to a particular development environment.

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 myUPS 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.

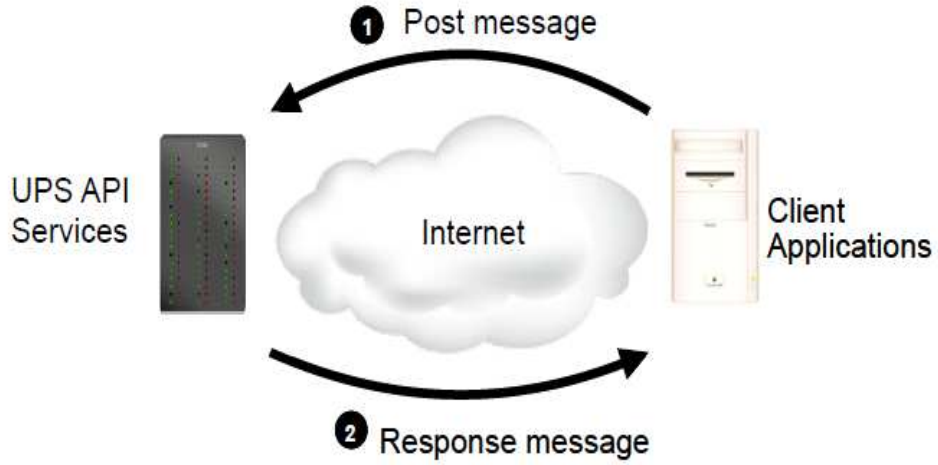


Figure 1. UPS Developer APIs accept requests from client applications in HTTP POST messages and reply to the requests with HTTP responses.

As part of the 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.

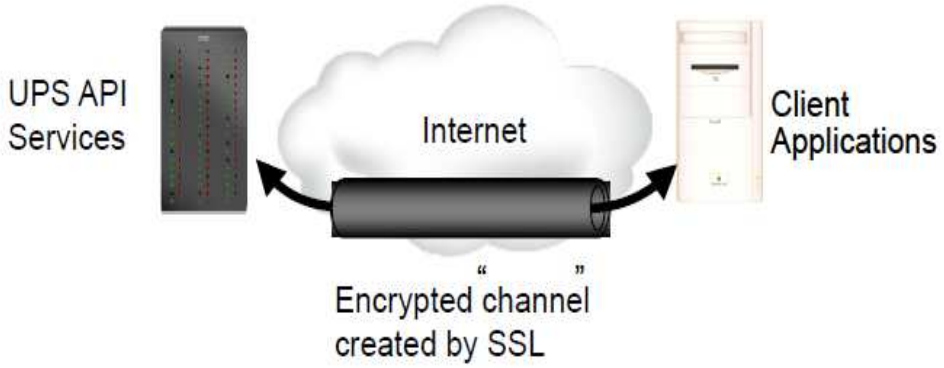


Figure 2: SSL creates a secure channel across a network and protects confidential communications using that channel.

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.

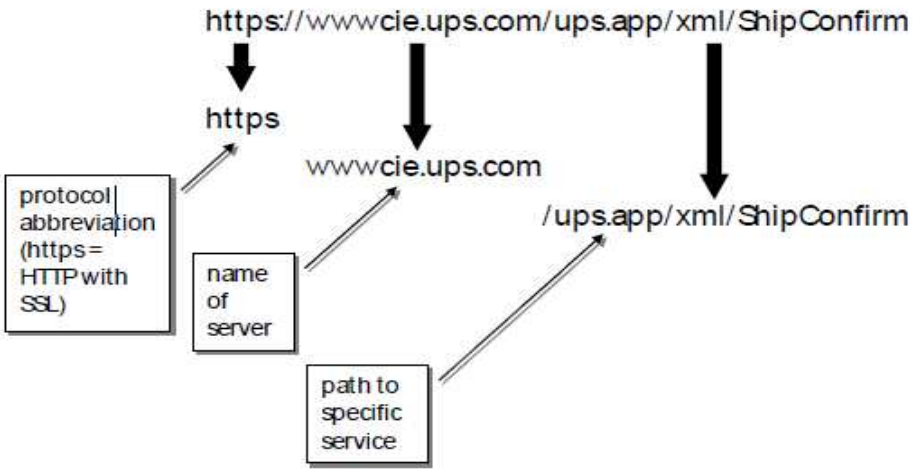


Figure 3. URLs identify communication protocols, servers, and specific services.

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");
HttpURLConnection conn = (HttpURLConnection) 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

1.6.2.4 SSL Certificate Changes and Renewals

UPS use the Chained Digital Certificates as a measure 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 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, https://onlinetools.ups.com/ups.app/xml/XAV, 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+hYxDANBgkqhkiG9w0BAQUFADCB
wTELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTwwOgYDVQQL
EzNDbGFzcyAzIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5
IC0gRzIxOjA4BgNVBAsTMShjKSAxOTk4IFZlcmlTaWduLCBJbmMuIC0gRm9yIGF1
dGhvcml6ZWQgdXNlIG9ubHkxHzAdBgNVBAsTFlZlcmlTaWduIFRydXN0IE5ldHdv
cmswHhcNMDkwMzI1MDAwMDAwWhcNMTkwMzI0MjM1OTU5WjCBtTELMAkGA1UEBhMC
VVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQLExZWZXJpU2lnbiBU
cnVzdCBOZXR3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2UgYXQgaHR0cHM6Ly93
d3cudmVyaXNpZ24uY29tL3JwYSAoYykwOTEvMC0GA1UEAxMmVmVyaVNpZ24gQ2xh
c3MgMyBTZWN1cmUgU2VydmVyIENBIC0gRzIwggEiMA0GCSqGSIb3DQEBAQUAA4IB
DwAwggEKAoIBAQDUVo9XOzcopkBj0pXVBXTatRlqltZxVy/iwDSMoJWzjOE3JPMu
7UNFBY6J1/raSrX4Po1Ox/lJUEU3QJ90qqBRVWHxYISJpZ6AjS+wIapFgsTPtBR/

RxUgKIKwaBLArlwH1/ZZzMtiVlxNSf8miKtUUTovStoOmOKJcrn892g8xB85essX
gfMMrQ/cYWIbEAsEHikYcV5iy0PevjG6cQIZTiapUdqMZGkD3pz9ff17Ybz8hHyI
XLTDe+1fK0YS8f0AAZqLW+mjBS6PLlve8xt4+GaRCMBeztWwNsrUqHugffkwer/4
3RlRKyC6/qfPoU6wZ/WAqiuDLtKOVImOHikLAgMBAAGjggKpMIICpTA0BggrBgEF
BQcBAQQoMCYwJAYIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLnZlcmlzaWduLm
NvbTAS
BgNVHRMBAf8ECDAGAQH/AgEAMHAGA1UdIARpMGcwZQYLYIZIAYb4RQEHF
wMwVjAo
BggrBgEFBQcCARYcaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL2NwczAqBggrBgEF
BQcCAjAeGhxodHRwczovL3d3dy52ZXJpc2lnbi5jb20vcnBhMDQGA1UdHwQtMCsw
KaAnoCWGI2h0dHA6Ly9jcmwudmVyaXNpZ24uY29tL3BjYTMtZzIuY3JsMA4GA1Ud
DwEB/wQEAwIBBjBtBggrBgEFBQcBDARhMF+hXaBbMFkwVzBVFglpbWFnZS9naW
Yw
ITAfMAcGBSsOAwIaBBSP5dMahqyNjmvDz4Bq1EgYLHsZLjAlFiNodHRwOi8vbG9n
by52ZXJpc2lnbi5jb20vdnNsb2dvLmdpZjApBgNVHREEIjAgpB4wHDEaMBgGA1UE
AxMRQ2xhc3MzQ0EyMDQ4LTEtNTIwHQYDVR0OBBYEFKXvCxHOwEEDo0plkEiy
HOBX
LX1HMIHnBgNVHSMEgd8wgdyhgcekgcQwgcExCzAJBgNVBAYTAlVTMRcwFQYDV
QQK
Ew5WZXJpU2lnbiwgSW5jLjE8MDoGA1UECxMzQ2xhc3MgMyBQdWJsaWMgUHJpb
WFy
eSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eSAtIEcyMTowOAYDVQQLEzEoYykgMTk
5
OCBWZXJpU2lnbiwgSW5jLiAtIEZvciBhdXRob3JpemVkIHVzZSBvbmx5MR8wHQYD
VQQLExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrghB92f4Hz6getxB5Z/uniTTGMA0G
CSqGSIb3DQEBBQUAA4GBAGN0Lz1Tqi+X7CYRZhr+8d5BJxnSf9jBHPniOFY6H5Cu
OcUgdav4bC1nHynCIdcUiGNLsJsnY5H48KMBJLb7j+M9AgtvVP7UzNvWhb98lR5e
YhHB2QmcQrmy1KotmDojYMyimvFu6M+O0Ro8XhnF15s1sAIjJOUFuNWI4+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

1.6.3 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
```

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.

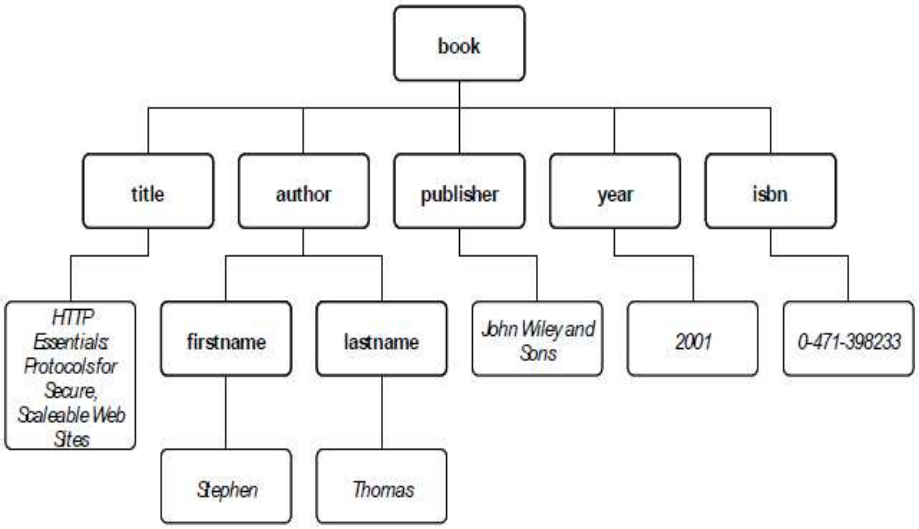


Figure 8. It is often convenient to show XML messages in a tree-like graph to highlight their structure.

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/.

1.6.4 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.

1.6.4.1 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 DOMDocument30
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 DOMDocument object.

Visual Basic 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

1.6.4.2 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

1.6.4.3 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 Street Level Services

With Address Validation Street Level web services, you can assist your customers by ensuring that addresses they provide are accurate. You can also use the API to determine if an address is a commercial or residential address. The API can help you reduce costly returns, provide better service to your customers, and more accurately determine shipping costs.

1.7.1 Address Validation

The Address Validation Street Level API check addresses against the United States Postal Service database of valid addresses in the US and Puerto Rico. If an address is not valid according to this database, the API can optionally provide a list of valid addresses that might correspond to the intended address.
Figure 1 shows how client applications use address validation services. The process begins when the client sends an AddressValidationRequest message to UPS API Services. UPS replies to this request with an AddressValidationResponse message.

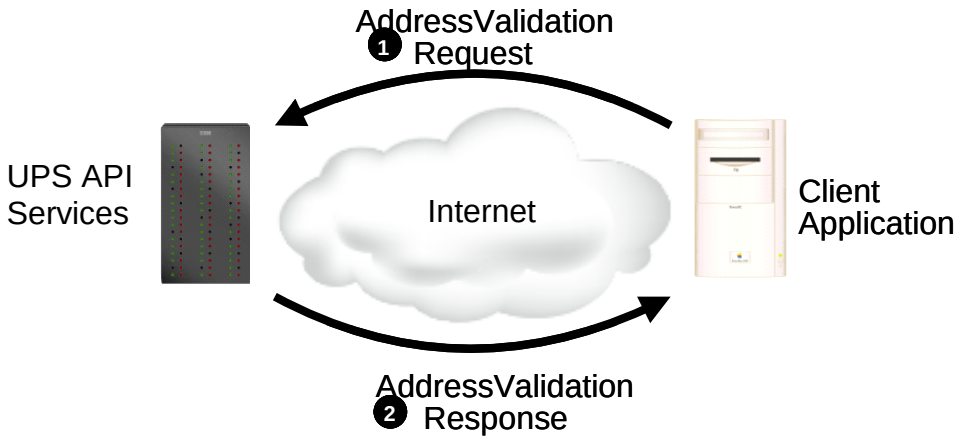


Figure 1. Address Validation Requires a Single Request-Response Exchange

Client applications can request either a general validation of a city, state, and zip code, or a validation of specific, street-level address. The RegionalRequestIndicator in the request determines which type of validation the application desires.
Client applications also indicate the maximum number of candidate addresses they wish to receive in the response. UPS returns candidate addresses only if the address that the client provides is not valid. Candidate addresses are valid addresses that might correspond to the requested address. If a client does not wish to receive candidate addresses, it can specify zero (0) for this value.

Example:

Validation of the SLAV is not at the suite/apt# level but, if a perfect match that contains a suite number or apt# is provided; the API will return a response that contains those elements.

Company or Name:	LAKESIDE PAIN CENTER
AddressLine:	6010 LAKESIDE COMMONS DR.
AddressLine:	STE B
City:	Macon
State:	GA
Zip:	31210
Country:	US

If the following address is sent in a request, the system will check if it exists. In this case, this address does not exist but the API will be able to return an address that maybe similar to the one that was provided as a candidate.

Company or Name:	LAKESIDE PAIN CENTER
AddressLine:	6010 LAKESIDE
AddressLine:	
City:	Macon
State:	
Zip:	312
Country:	US

The response for a request with the address provided below will illustrate the fact that in order to get a correct address including the suite/apt #; those elements must be provided accurately or the address must be distort completely for the system to perform a thorough search.

Company or Name:	LAKESIDE PAIN CENTER
AddressLine:	6010 LAKESIDE COMMONS DR.
AddressLine:	STE X
City:	Macon
State:	GA
Zip:	31210
Country:	US

1.7.2 Address Classification

The Address Validation Street Level API can also determine whether a given address is a residential or commercial address. Address classification uses the same request/response exchange as address validation. In fact, classification and validation can be combined in a single request.

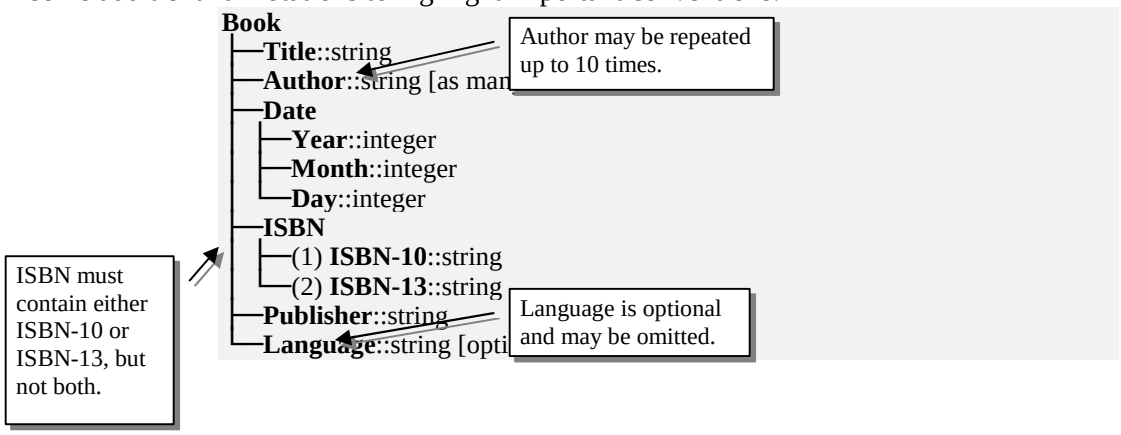
For address classification, UPS determines whether an address is a residence or a commercial location. For address classification requests, it is important that the user include as much information as possible or available about the address, for example, a contact name or "attention to" value. Such information is important for accurate results, as many locations include both commercial and residential entities (such as a deli on the ground floor of an apartment building).

1.8 The Address Validation Street Level 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.

1.8.1 Describing UPS API Services 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.

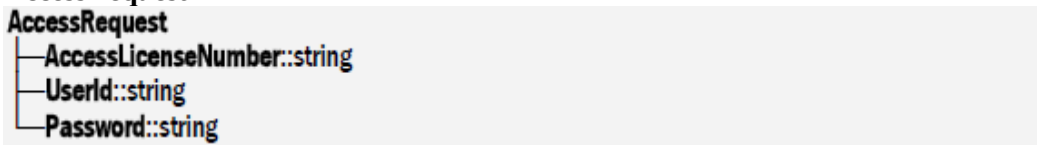
Finally, notice that the children of the ISBN element (ISBN-10 and ISBN-13) have a number in parentheses preceding them. This number indicates a choice. In this example, the ISBN element must have either an ISBN-10 child element or an ISBN-13 child element, but it cannot have both.

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.

1.8.2 Authenticating Client Applications

UPS 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>
```

Example AddressValidationRequest Message

```
<?xml version="1.0" ?>
<AccessRequest xml:lang='en-US'>
    <AccessLicenseNumber>YOURACCESSLICENSENUMBER</AccessLicenseNumber>
    <UserId>YOURUSERID</UserId>
    <Password>YOURPASSWORD</Password>
</AccessRequest>
<?xml version="1.0" ?>
<AddressValidationRequest xml:lang='en-US'>
    <Request>
        <TransactionReference>
            <CustomerContext /><XpciVersion>1.0001</XpciVersion>
        </TransactionReference>
        <RequestAction>XAV</RequestAction>
        <RequestOption>3</RequestOption>
    </Request>
    <MaximumListSize>3</MaximumListSize>
    <AddressKeyFormat>
        <ConsigneeName>CONSIGNEE NAME</ConsigneeName>
        <BuildingName>BUILDING NAME</BuildingName>
```

```
            <AddressLine>ADDRESS LINE 1</AddressLine>
            <AddressLine>ADDRESS LINE 2</AddressLine>
            <AddressLine>ADDRESS LINE 3</AddressLine>
            <PoliticalDivision2>POLITICAL DIVISION</PoliticalDivision2>
            <PoliticalDivision1>XX</PoliticalDivision1>
            <PostcodePrimaryLow>00000</PostcodePrimaryLow>
            <CountryCode>YY</CountryCode>
        </AddressKeyFormat>
    </AddressValidationRequest>
```

Example AddressValidationResponse Message

```
    <?xml version="1.0" ?>
    <AddressValidationResponse>
        <Response>

    <TransactionReference><XpciVersion>1.0001</XpciVersion></TransactionRe
    ference>
            <ResponseStatusCode>1</ResponseStatusCode>
            <ResponseStatusDescription>Success</ResponseStatusDescription>
        </Response>
        <AmbiguousAddressIndicator />
        <AddressClassification><Code>2</Code></AddressClassification>
        <AddressKeyFormat>
            <AddressClassification><Code>2</Code></AddressClassification>
            <ConsigneeName>CONSIGNEE NAME</ConsigneeName>
            <AddressLine>ADDRESS LINE</AddressLine>
            <PoliticalDivision2>POLITICAL DIVISION</PoliticalDivision2>
            <PoliticalDivision1>XX</PoliticalDivision1>
            <PostcodePrimaryLow>00000</PostcodePrimaryLow>
            <PostcodeExtendedLow>0000</PostcodeExtendedLow>
            <CountryCode>YY</CountryCode>
        </AddressKeyFormat>
        <AddressKeyFormat>
            <AddressClassification><Code>1</Code></AddressClassification>
            <AddressLine>ADDRESS LINE</AddressLine>
            <PoliticalDivision2>POLITICAL DIVISION</PoliticalDivision2>
            <PoliticalDivision1>YY</PoliticalDivision1>
            <PostcodePrimaryLow>00000</PostcodePrimaryLow>
            <PostcodeExtendedLow>0000</PostcodeExtendedLow>
            <CountryCode>YY</CountryCode>
        </AddressKeyFormat>
    </AddressValidationResponse>
```

1.8.3 XAV Request

Address Validation Request

The AddressValidationRequest message consists of two complete XML documents. The first document is an AccessRequest. It is immediately followed by an AddressValidationRequest. 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/XAV

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

https://onlinetools.ups.com/ups.app/xml/XAV

The AddressValidationRequest document in the request must conform to the following XML structure.

Name	XPath	Required	Max Allowed	Type	Length	Description
AddressValidation Request	/AddressValidationRequest	Yes	One	Container	N/A	N/A
Request	/AddressValidationRequest/Request	Yes	One	Container	N/A	N/A
TransactionReference	/AddressValidationRequest/Request/TransactionReference	No	One	Container	N/A	TransactionReference identifies transactions between client and server.
CustomerContext	/AddressValidationRequest/Request/TransactionReference/CustomerContext	No	One	string	1...512	The client uses CustomerContext to synchronize request/response pairs. The client establishes CustomerContext, which can contain any information you want, as along as it is valid XML; it is echoed back by the server.
RequestAction	/AddressValidationRequest/Request/RequestAction	Yes	One	string	3	Indicates the action to be taken by the XML service. Must be 'XAV'.
RequestOption	/AddressValidationRequest/Request/ RequestOption	Yes	One	string	1	Identifies the optional processing to be performed. If not present or invalid value then an error will be sent back. Valid Values are: 1 - Address Validation 2 - Address Classification 3 - Address Validation and Address Classification. Note: Please see Country Validation and Classification Matrix for valid request combinations
RegionalRequestIndicator	/AddressValidationRequest/RegionalRequestIndicator	No	One	EMPTY		If this indicator is present then either the region element or any combination of Political Division 1, Political Division 2, PostcodePrimaryLow and the PostcodeExtendedLow fields will be recognized for validation in addition to the urbanization element. If this tag is present, US and PR street level address validation will not occur. The default is to provide street level address validation. Not valid with the address classification request option.

Name	XPath	Required	Max Allowed	Type	Length	Description
MaximumListSize	/AddressValidationRequest/MaximumListSize	No	One	Container	1...3	The maximum number of Candidates to return for this request. Valid Values: 0-50 Default value is 15
AddressKeyFormat	/AddressValidationRequest/AddressKeyFormat	Yes	One	Container		The Key format is based on addressing standards jointly developed by the Postal Service and mailing industry. The information provided in the Address Key container will be returned in the same format.
ConsigneeName	/AddressValidationRequest/AddressKeyFormat/ConsigneeName	No	One	string	1...40	Name of business, company or person. Ignored if user selects the RegionalRequestIndicator.
BuildingName	/AddressValidationRequest/AddressKeyFormat/BuildingName	No	One	string	1...40	Name of building. Ignored if user selects the RegionalRequestIndicator.
AddressLine	/AddressValidationRequest/AddressKeyFormat/AddressLine	No	Three	string	1...100	Address line (street number, street name and street type) used for street level information. Applicable to US and PR only. Ignored if user selects the RegionalRequestIndicator.
Region	/AddressValidationRequest/AddressKeyFormat/Region	No	One	string	1...100	Single entry containing in this order Political Division 2, Political Division 1, Post Code Primary Low, and Post Code Extended Low. If the node is present the following tags will be ignored Political Division 2, Political Division 1, Post Code Primary Low, and Post Code Extended Low. Valid only for US or PR origins only. Using this tag for non US/PR origins may cause address format errors.
PoliticalDivision2	/AddressValidationRequest/AddressKeyFormat/PoliticalDivision2	No	One	string	1...30	City or town name.
PoliticalDivision1	/AddressValidationRequest/AddressKeyFormat/PoliticalDivision1	No	One	string	1...30	State or Province/Territory name.
PostcodePrimaryLow	/AddressValidationRequest/AddressKeyFormat/PostcodePrimaryLow	No	One	string	1...10	Postal Code.

Name	XPath	Required	Max Allowed	Type	Length	Description
PostcodeExtended Low	/AddressValidationRequest/AddressKeyFormat /PostcodeExtendedLow	No	One	string	1...10	4 digit Postal Code extension. For U.S. use only.
Urbanization	/AddressValidationRequest/AddressKeyFormat /Urbanization	No	One	string	1...30	Political Division 3. Only Valid for Puerto Rico.
CountryCode	/AddressValidationRequest/AddressKeyFormat /CountryCode	Yes	One	string	2	Country Code. Currently the only valid values are US, PR, or CA. Note: Please see country validation and Classification Matrix for valid request combinations

1.8.4 XAV Response

The 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	1	Container	N/A	N/A
Response	/AddressValidationResponse/Response	Yes	One	Container	N/A	N/A
TransactionReference	/AddressValidationResponse/Response/TransactionReference	Yes	One	Container	N/A	N/A
CustomerContext	/AddressValidationResponse/Response/TransactionReference/CustomerContext	No	One	String	1...512	Echoes back the Customercontext from the Request
XpciVersion	/AddressValidationResponse/Response/TransactionReference/XpciVersion	No	One	String	1...15	Represents the version of the schema used by the XAV Tool
ResponseStatusCode	/AddressValidationResponse/Response/ResponseStatusCode	Yes	One	String	1	Identifies the success or failure of the transaction. 1 = Success 0 = Failure
ResponseStatusDescription	/AddressValidationResponse/Response/ResponseStatusDescription	No	One	String	1...35	Describes Response Status Code. Returns text of 'Success' or 'Failure'.
Error	/AddressValidationResponse/Response/Error	Yes	One	Container	N/A	Error Container. The error is described with a code and description.
ErrorSeverity	/AddressValidationResponse/Response/Error/ErrorSeverity	Yes	One	String	1...10	Valid values: 'Hard' - The provided Request data will always fail 'Transient' - A UPS application is suffering an intermittent failure. Try the transaction again later.

Name	XPath	Required	Max Allowed	Type	Length	Description
ErrorCode	/AddressValidationResponse/Response/Error/ErrorCode	Yes	One	String	1..2	Warning code returned by the system.
ErrorDescription	/AddressValidationResponse/Response/Error/ErrorDescription	No	One	String	1...150	
MinimumRetrySeconds	/AddressValidationResponse/Response/Error/MinimumRetrySeconds	No	One	String	1...15	How long to wait before re-transmitting a Transaction that failed with a 'Transitive' error
ErrorLocation	/AddressValidationResponse/Response/Error/ErrorLocation	Yes	One	Container	1...150	XPATH of the element causing the 'Hard' error
ErrorLocationElementName	/AddressValidationResponse/Response/Error/ErrorLocation/ErrorLocationElementName	No	One	String	1...30	Element whose value causes the 'Hard' error
ErrorLocationAttributeName	/AddressValidationResponse/Response/Error/ErrorLocation/ErrorLocationAttributeName	No	One	String		
ErrorDigest	/AddressValidationResponse/Response/Error/ErrorDigest	No	One	String	Unbounded	Details of the error
ValidAddressIndicator	/AddressValidationResponse/ValidAddressIndicator	Cond	One	EMPTY	N/A	The input address matches an address in the Address Matching Systems database; under the condition where the Address Matching System is able to explicitly differentiate an address from all others in the database.
AmbiguousAddressIndicator	/AddressValidationResponse/AmbiguousAddressIndicator	Cond	One	EMPTY	N/A	The Address Matching System is not able to explicitly differentiate an address from any other one in the database. The list of potential matching addresses may be included in the Candidate list.

Name	XPath	Required	Max Allowed	Type	Length	Description
NoCandidatesIndicator	/AddressValidationResponse/NoCandidatesIndicator	Cond	One	EMPTY	N/A	The Address Matching System is not able to match an address from any other one in the database.
AddressClassification	/AddressValidationResponse/AddressClassification	No	One	Container	N/A	Container returning the classification of the input address, if requested
Code	/AddressValidationResponse/AddressClassification/Code	Yes*	One	string	1	Contains the classification code of the input address. 0 = Unknown 1= Commercial 2 = Residential
Description	/AddressValidationResponse/AddressClassification/Description	Yes*	One	string	1...15	Contains the text description of the address classification code: Unknown Commercial Residential
AddressKeyFormat	/AddressValidationResponse/AddressKeyFormat	Cond	One	Container	N/A	The Key format is based on addressing standards jointly developed by the Postal Service and mailing industry. The information provided in the Address Key container will be returned in the same format.
AddressClassification	/AddressValidationResponse/AddressKeyFormat/AddressClassification	No	One	Container	N/A	Container returning the classification of the address, if requested
Code	/AddressValidationResponse/AddressKeyFormat/AddressClassification/Code	Yes*	One	string	1	Contains the classification code of the address. 0 = Unknown 1 = Commercial 2 = Residential
Description	/AddressValidationResponse/AddressKeyFormat/AddressClassification/Description	Yes*	One	string	1...15	Contains the text description of the address classification code. Unknown Commercial Residential
ConsigneeName	/AddressValidationResponse/AddressKeyFormat/ConsigneeName	No	One	string	1...40	Name of business, company or person. Not returned if user selects the RegionalRequestIndicator.

Name	XPath	Required	Max Allowed	Type	Length	Description
BuildingName	/AddressValidationResponse/AddressKeyFormat/BuildingName	Cond	One	string	1...40	Name of building. Not returned if user selects the RegionalRequestIndicator.
AddressLine	/AddressValidationResponse/AddressKeyFormat/AddressLine	Cond	Three	string	1...100	Address line (street number, street name and street type, and political division 1, political division 2 and postal code) used for street level information. Applicable to US and PR only. Not returned if user selects the RegionalRequestIndicator.
PoliticalDivision2	/AddressValidationResponse/AddressKeyFormat/PoliticalDivision2	Cond	One	string	1...30	City or Town.
PoliticalDivision1	/AddressValidationResponse/AddressKeyFormat/PoliticalDivision1	Cond	One	string	1...30	State/Province. Returned if the location is within a State/Province/Territory. For International: returned if user enters valid Country Code, and City/postal code and it has a match. For Domestic addresses, the value must be a valid 2-character value (per US Mail standards).For International the full State or Province name will be returned.
PostcodePrimaryLow	/AddressValidationResponse/AddressKeyFormat/PostcodePrimaryLow	Cond	One	string	1...10	Low-end Postal Code. Returned for countries with Postal Codes. May be alphanumeric
PostcodeExtendedLow	/AddressValidationResponse/AddressKeyFormat/PostcodeExtendedLow	No	One	string	1...10	Low-end extended postal code in a range. Example in quotes: Postal Code 30076-'1234'. Only returned in candidate list. May be alphanumeric
Region	/AddressValidationResponse/AddressKeyFormat/Region	No	One	string	1...30	Single entry containing in this order Political Division 2, Political Division 1 and Post Code Primary Low and/or PostcodeExtendedLow.
Urbanization	/AddressValidationResponse/AddressKeyFormat/Urbanization	No	One	string	1...10	Puerto Rico Political Division 3. Only Valid for Puerto Rico.
CountryCode	/AddressValidationResponse/AddressKeyFormat/CountryCode	Yes	One	string	1...2	A country code. Required to be returned.

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 Street Level Address Validation

Test your Street Level Address Validation application with valid and invalid address elements.

Note: In the Customer Integration Environment, Street Level Address Validation will only produce results for addresses in New York (NY) and California (CA).

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 Street Level Address Validation **XML Tool** software to https://wwwcie.ups.com/ups.app/xml/XAV.

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://wwwcie.ups.com/ups.app/xml/XAV

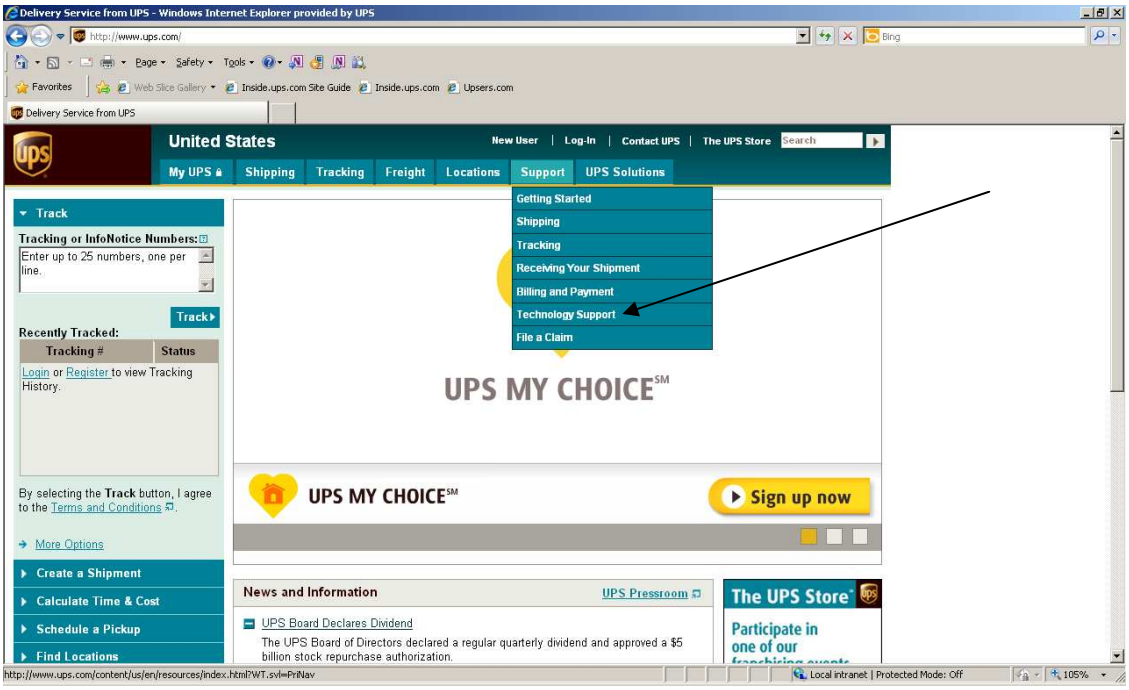
You should see the following in the browser window:
Service Name: XAV
Remote User: null
Server Port: 443
Server Name: wwwcie.ups.com
Servlet Path: /XAV

Once testing is completed please direct your Address Validation Street Level XML software to **https://onlinetools.ups.com/ups.app/xml/XAV**

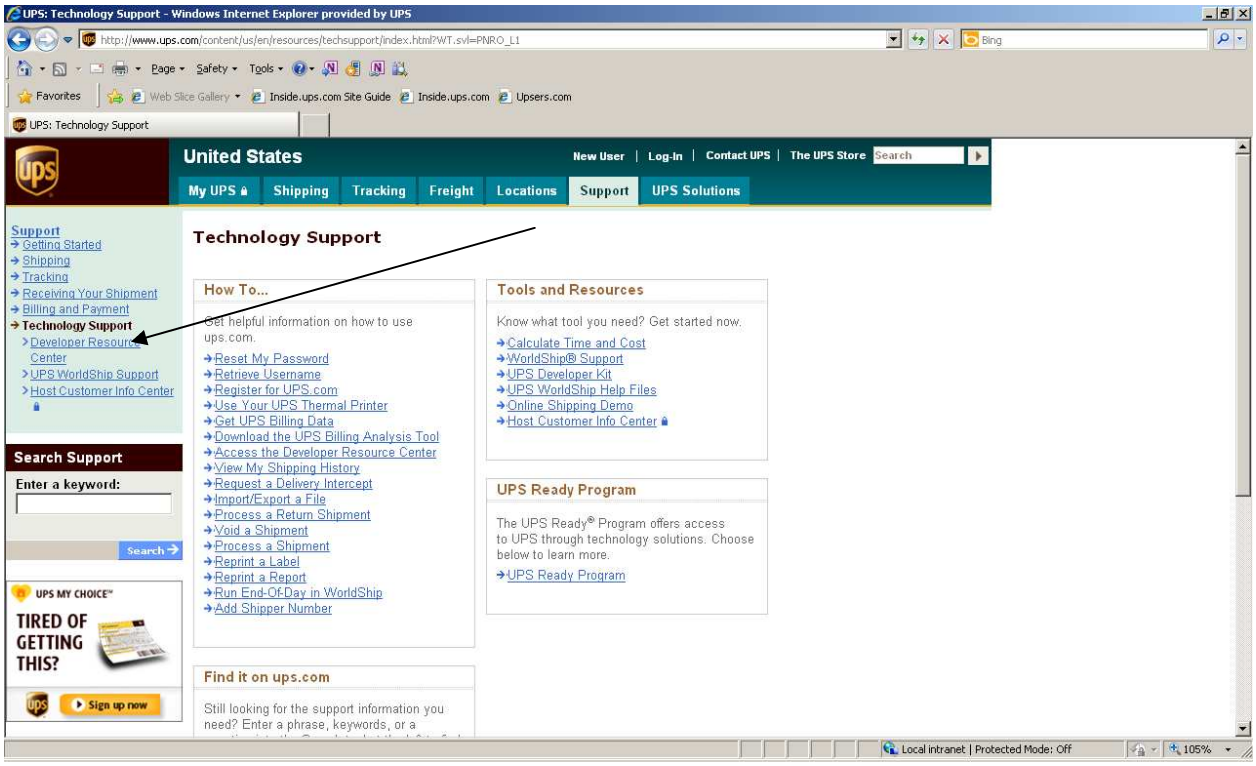
Appendix A - Accessing Tech Support (US Site Example)

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

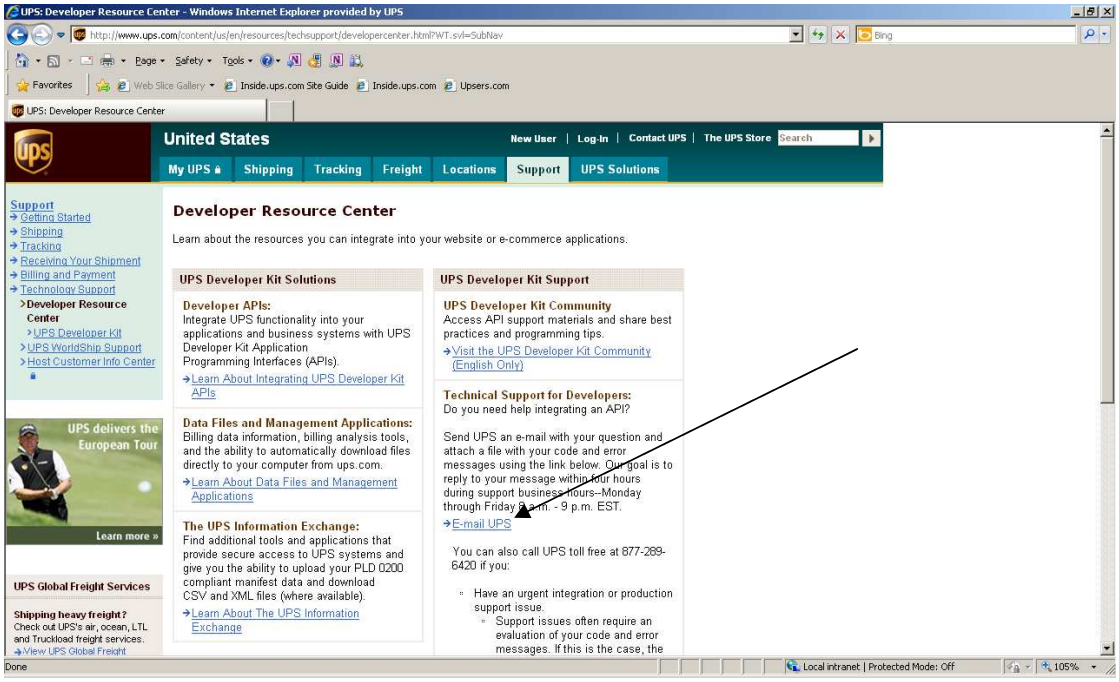
Hover over the “Support” tab, and select “Technology Support” from the drop-down.



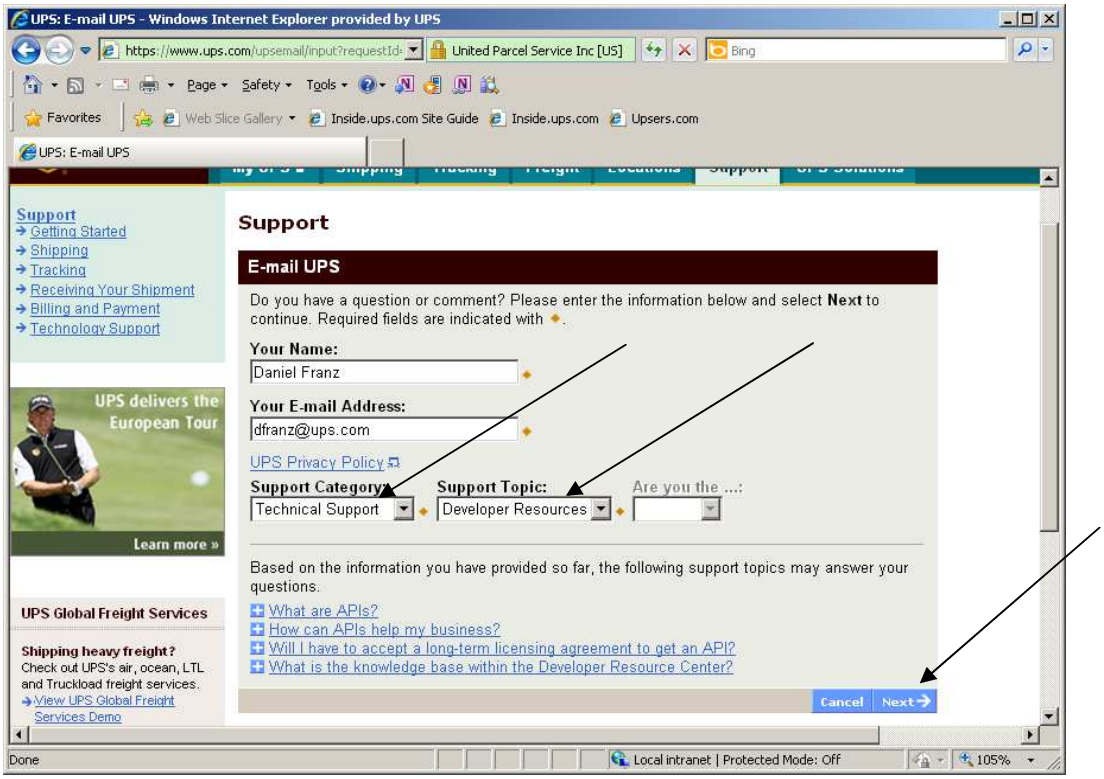
From the Technology Support page select the Developer Resource Center link from the left navigation.



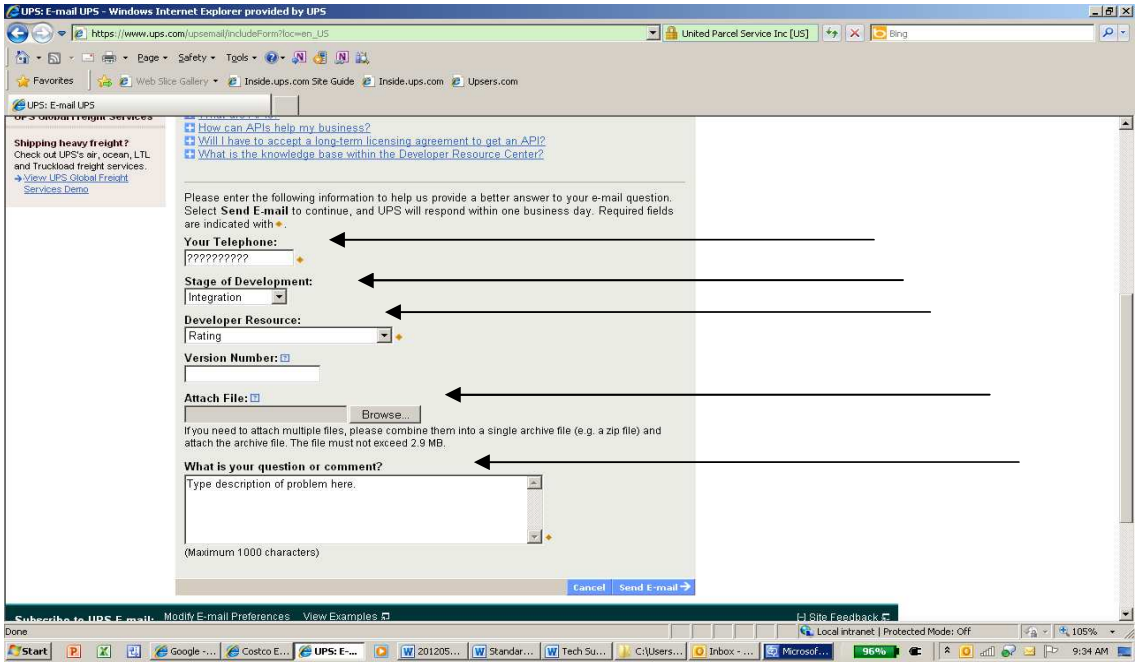
From the Developer Resource Center page select the “Email UPS” link in right hand column.



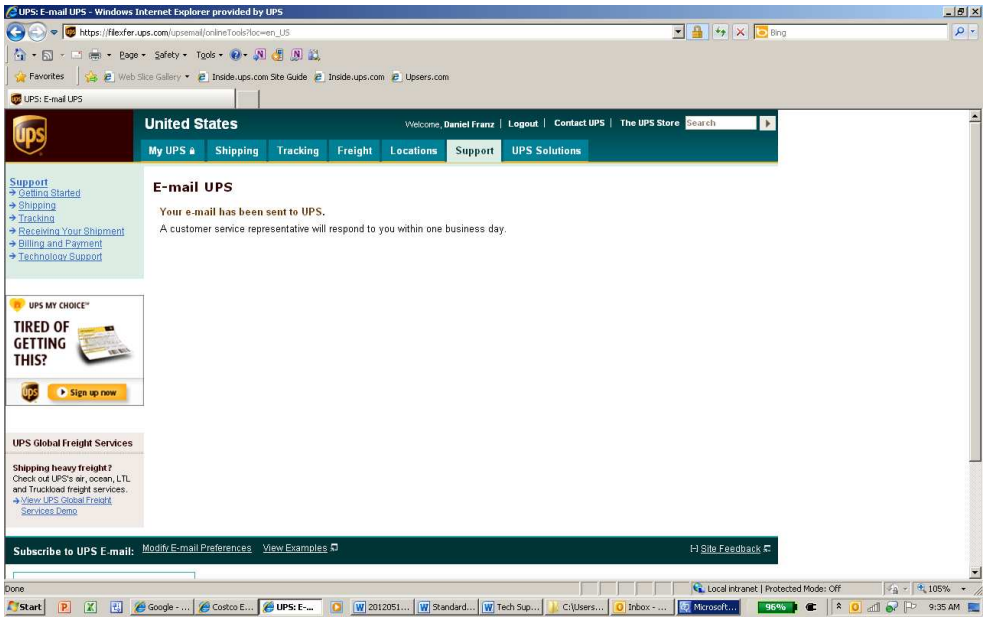
Complete Name, email address, Support Category must be “Technical Support”, and Support Topic must be “Developer Resource”. Select the “Next” button.



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.



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



Appendix B – Address Validation Street Level 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
260050	Hard	Invalid Request Action	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.	The country code is not US or PR.
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.
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.

Error Code	Severity	Description	Condition
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.
264030	Hard	The state is not supported in the Customer Integration Environment.	

XML Errors

Error Code	Severity	Description
10001	Hard	The XML document is not well formed
10002	Hard	The XML document is well formed but the document is not valid
10003	Hard	The XML document is either empty or null
10006	Hard	Although the document is well formed and valid, the element content contains values which do not conform to the rules and constraints contained in this specification
10013	Hard	The message is too large to be processed by the Application

Common Architecture Errors

Error Code	Severity	Description
20001	Hard	General process failure
20002	Hard	The specified service name, {0}, and version number, {1}, combination is invalid
20003	Hard	Please check the server environment for the proper J2EE ws apis

Common License and Access Control Errors

Error Code	Severity	Description
250000	Hard	No XML declaration in the XML document
250001	Hard	Invalid Access License for the tool. Please re-license.
250002	Hard	Invalid UserId/Password
250003	Hard	Invalid Access License number
250004	Hard	Incorrect UserId or Password
250005	Hard	No Access and Authentication Credentials provided
250006	Hard	The maximum number of user access attempts was exceeded
250007	Hard	The UserId is currently locked out, please try again in 24 hours
250009	Hard	License Number not found in the UPS database

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	Peurto 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)		
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 - Country Classification and Validation Matrix

The correct combinations for Country Classification and Validation are listed below. The classification determines whether a given address is residential or commercial.
A Country Classification is returned for the countries indicated below.
A Street Level Address Validation is returned for the countries indicated below.

Country	Classification	Validation
US	X	X
PR		X
CA	X	

Appendix E - Frequently Asked Questions: Address Validation Street Level

API	Category	Question	Answer
Address Validation Street Level	General	What countries' addresses can be validated by the Address Validation - Street Level API?	The Address Validation - Street Level API allows the validation of street level address in the US and Puerto Rico only. [Note: AVSL supports classification for US & Canadian addresses only.]
Address Validation Street Level	General	Does the Address Validation API classify addresses?	The Address Validation API does not classify addresses. The Address Validation - Street Level API classifies addresses in both US and Canada. The API classifies addresses as Residential or Commercial in keeping with UPS standards.
Address Validation Street Level	General	Why do we get back a candidate list of addresses even when the response has a Valid Address Indicator?	The API returns a candidate list even when there is a Valid Address Indicator because the parameters entered have more then one valid match. If the address that was entered as part of the request is returned as part of the Candidate list then that address is valid and should be used.
Address Validation Street Level	General	How is street level AV completed? Does it use the USPS to validate an address?	Currently the Address Validation Street Level API's database is updated at monthly intervals with new address information from USPS. Generally the database update will occur around the 15th of the month. The actual date changes from month to month dependent upon the amount of testing a given data set might require issues that are found, and other factors that can contribute to the swiftness of data validation. The database updates should synchronize the information with the USPS. Any addresses not currently available through UPS will be added in the following month.
Address Validation Street Level	General	How much time will it take to program/implement the Address Validation - Street Level API?	The programming/implementation of 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 Street Level	Address Validation frequency of updates	How frequently are the databases updated for Address Validation Street Level?	Address Validation Street Level API 1. Classification = weekly on Sunday input from operations. 2. Validation = monthly day varies.
Address Validation Street Level	Batch Upload	Does either the Address Validation or Address Validation Street Level APIs offer the ability to batch	No. Neither API provides batch upload. Only individual requests.

API	Category	Question	Answer
		upload?	
Address Validation Street Level	Suite/Apt #	Does Address Validation Street Level API provide a candidate list for addresses that have suite/apt information?	The API does not return candidate lists for suite or apartment number ranges.
Address Validation Street Level	CASS Certified	Is the Address Validation Street Level API CASS certified or the data we receive CASS certified?	UPS doesn't provide any API or application with customer-facing CASS-certified address validation. CASS certification is required for discounts on USPS services, but not required for any UPS services or discounts. UPS address validation is not CASS-certified; it's also free as we are not competing with vendors that provide CASS-certified Address Validation software. The Address Validation API is provided to allow customers who don't already use CASS-certified Address Validation software, our Address Validation Street Level API can be used to help clean-up addresses for UPS shipments. UPS delivers to addresses that are not in the USPS database (some examples are addresses that are warehouses that don't accept mail and areas where the USPS only provides PO Box delivery) - so an invalid address may still be deliverable. Also, a valid address may be the wrong address, and still require an address correction. NOTE: The source of our data for address validation is the USPS that we subscribe to and refresh monthly. So for customers who do scrub their addresses with CASS-certified Address Validation software, the Address Validation API provided by UPS doesn't provide any additional benefit.
Address Validation Street Level	Resi/Comm Indicator	Which APIs provide address classification?	The Address Validation Street Level API provides address classification only.

API	Category	Question	Answer
Address Validation Street Level	Ambiguous Address Indicator	How do we determine whether an address is "ambiguous" or not?	Ambiguous Address Indicator = returned when the address validation score for the entire address is below a UPS specified confidence threshold. Address Validation Score = average of detailed validation score and regional validation score. Detailed Validation Score = street number information validated based upon USPS feed using proprietary UPS validation database. Regional Validation Score = city, state, zip validated based upon USPS feed using proprietary UPS validation database. UPS address information is updated monthly through a subscription with USPS. In the case of the address you specified... "2724 S. PECK ROAD 91016". <u>Detail portion</u> 2724 S. PECK ROAD gets a grade of B 2724 PECK ROAD gets a grade of A <u>Regional portion</u> 91016 gets a grade of C Monrovia CA 91016 gets a grade of A When the two scores were averaged the score was below the threshold and therefore the ambiguous address indicator returned. In this scenario, had you removed the "S." with the same information you provided the validation score average would have been high enough and the Valid Address Indicator returned. Also, had you provided the city & state for the regional portion then the score on the regional side would have been high enough that when we averaged the two it would have raised the average over the threshold, even though the detailed portion had the "S." included, and a Valid Address Indicator would have been returned.
Address Validation Street Level	Batch Upload	Does the Address Validation Street Level API offer the ability to batch upload?	No. The API does not allow batch upload but only individual requests.
Address Validation Street Level	Resi/Comm Database	What process does UPS use to apply residential/commercial designations?	Driver classification at delivery is used to create a database. All tools, all shipping applications, all Driver classifications, all center audits, and all Billing access the same database. The information entered by the driver for changes is audited and edited by the center team prior to changes to the database being entered.
Address Validation Street Level	Valid/Invalid Address	How do I know whether the address I submitted is correct?	If the address is entered incorrectly the API will return an AmbiguousAddressIndicator and a list of candidate addresses. If the address is entered correctly the API will return a ValidAddressIndicator and the correct address.

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 chain 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	General XML - Coding - Limited XML Coding Knowledge	I have no XML coding knowledge do you have any recommendations for how to get started?	Thank you for your inquiry. The only direct example of the XML APIs implemented is the Brown and Brown store whose demo can be viewed by clicking the following link and viewing the PDF document: http://www.ups.com/content/us/en/bussol/offering/online_tools/downloads.html. To become more familiar with XML we would have to recommend reviewing some of the references listed in the Bibliography section at the end of the Rating API - Package Developer's Guide. There are also examples of XML in both the Developer's and Reference Guides. The Xpath in the Developer's Guide lists the descriptions of the XML tags and potential values. Once one is familiar with XML, the Xpath will help you fashion specific requests which can be posted to our servers and receive a proper XML response. The "readme.html" document in the Developer's Kit will help you get started using the two guides. Please note that the APIs were designed to be implemented by customers with a prior knowledge of XML along with a programming language. The main programming requirements the UPS Developer APIs have is that the XML document must be submitted to our servers via a POST using HTTPS (SSL). Depending upon your business needs and programming knowledge, your application can be simple or elaborate. Unfortunately, we do not support code directly; however, Java and Visual Basic code examples are supplied in the Reference Guide which can help guide customers to develop their own application. These examples are for a theoretical implementation and one may use portions of the code provided based on their environment and their application's design. If you are still deciding if the UPS Developer APIs will fulfill your business needs, you may also want to contact your UPS Account Representative for further guidance.

API	Category	Question	Answer
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.
All	Technical Support email form	From where XPCI version number obtained is 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.

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.
All	Web Services - Empty folders within the documentation zip file.	The ship_developer's 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: <ErrorLocationElementName>XPATH TO FIRST ELEMENT WHERE XML DOESN'T MATCH EXPECTED FORMAT</ErrorLocationElementName> 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, customer's 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 ESTSa-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. If they must see a visual, Daniel Franz, the Online Tools product manager refers customers to a UPS Ready provider that actually shows a demo of our tools. This may be better than just a screenshot from one customer's web site. Interactive example at shipworks.com http://www.interapptive.com/shipworks/quicktour.html Also, UPS is a user of our tools. Our Calculate Time and Cost application uses our Rating and Time in Transit Tools; Internet Shipping using our Shipping Tool; and Tracking at ups.com uses our Tracking Tool.
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.

