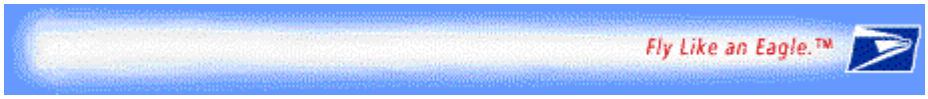
United States Postal Service
Web Tool Kit Development Guide



Development Guide for

Application Programming Interfaces to USPS Web Tools Service

Information and Procedures Prior to Use

Document Version 4.4 (02/08/2018)

To Our Customers

In the e-mail from the registration process, you received a user ID that will ultimately allow you to utilize the Web Tools APIs. Any additional documentation or contact with you will be made through the contact person indicated on the registration form.

If you require technical support, contact the USPS Internet Customer Care Center (ICCC). This office is staffed as follows:

- Monday through Friday from 8:00 a.m. to 8:30 p.m. Eastern Time
- Saturdays from 8:00 a.m. to 6:00 p.m. Eastern Time
- Sunday and Postal Holidays - Closed except for the following Holidays: Martin Luther King; President's Day; Columbus Day; & Veteran's Day with hours from 9:00 a.m. to 6:00 p.m. ET.

E-mail address: uspstechnicalsupport@mailps.custhelp.com

Telephone: 1-800-344-7779

USPS Customer Commitment

The United States Postal Service fully understands the importance of providing information and service anytime day or night to your Internet and e-commerce customers. For that reason, the USPS is committed to providing 24 x 7 service from our API servers, 365 days a year.

Thank you for helping the U.S. Postal Service provide new Internet services to our shipping customers.

Internet Shipping Solutions Team
U.S. Postal Service
475 L'Enfant Plaza, SW
Washington, DC 20260-2464

Registered Trademarks

Priority Mail, Priority Mail Flat Rate, Priority Mail International, Priority Mail Regional Rate, Global Express Mail, Global Express Guaranteed, Global Priority Mail, Parcel Post, Parcel Select, First-Class Mail, USPS, USPS Web Tools, and ZIP + 4 are registered trademarks of the U.S. Postal Service.

Priority Mail Express, Priority Mail Express 1-Day, Priority Mail Express 2-Day, Priority Mail Express 3-Day, Priority Mail Express DPO, Priority Mail Express International, Priority Mail Express Intl, Priority Mail Express Military, Priority Mail Express Offshore, Priority Mail 1-Day, Priority Mail 2-Day, Priority Mail 3-Day, Priority Mail DPO, Priority Mail Intl, Priority Mail Military, Priority Mail Offshore, Signature Confirmation, Standard Post, USPS Tracking, ZIP, and ZIP Code are trademarks of the U.S. Postal Service.

Microsoft and Visual Basic are registered trademarks of Microsoft Corporation.

Adobe Acrobat and Adobe Reader are trademarks of Adobe Systems Incorporated.

DUNS is a registered trademark of Dun & Bradstreet.

Table of Contents

Introduction

The USPS Web Tool Kit Application Program Interfaces (APIs) allow developers of web-based and shrink-wrapped applications access to the online services of the United States Postal Service (USPS). They provide easy access to shipping information and services for your customers. By integrating these APIs into your web site, your customers can utilize the functions provided by the USPS without ever leaving your web site. Once the APIs are integrated into your web site, the USPS Shipping API Server communicates over HTTP using XML (Extensible Markup Language).

Implementing these APIs requires experienced programmers who are familiar with Internet and web site development tools and techniques.

This document provides guidance and step-by-step instructions for installing the USPS APIs, as well as fulfilling various administrative requirements. The administrative requirements vary between different APIs (e.g., submitting samples of labels for some APIs, signing a licensing agreement for certain software, etc.), and this document provides guidance to navigate through the process for the API you are implementing. *It is imperative that developers read this manual first, as it provides necessary information and procedures prior to use.*

There is a *Web Tool Kit User's Guide* for each API available at http://www.usps.com/webtools/. These user guides provide information of the XML transactions to the USPS Shipping API server.

User ID Restrictions

The user ID that you have received is for you or your company to use in accordance with the Terms and Conditions of Use to which you agreed during the registration process. *This user ID is not to be shared with others outside your organization, nor is it to be packaged, distributed, or sold to any other person or entity.* Please refer to the Terms and Conditions of Use Agreement for additional restrictions on the use of your user ID, as well as this document and the APIs contained herein.

Warning: If the U.S. Postal Service discovers use of the same user ID from more than one web site, all users will be subject to immediate loss of access to the USPS server and termination of the licenses granted under the Terms and Conditions of Use.

The documentation and sample code contained in the *Web Tool Kit User Guide* series may be reused and/or distributed to your customers or affiliates to generate awareness, encourage Web Tool use, or provide ease-of-use. However, it is your responsibility to ensure that your customers do not use your user ID. Direct your users to www.usps.com/webtools/ so that they can register, agree to the Terms and Conditions of Use agreement, and receive their own unique user ID.

Note to Software Distributors: The User ID restrictions discussed above are intended for e-tailers that use the USPS Web Tools exclusively within their own web sites. If you plan to distribute software with the USPS Web Tools embedded, you must refer to the "Software Developers' Terms and Conditions of Use" available at http://www.usps.com/webtools/.

For more information regarding the USPS Web Tool Kit user ID policy, or for questions regarding the distribution of documentation, please send an e-mail to the Internet Customer Care Center (ICCC) at uspstechnicalsupport@mailps.custhelp.com.

USPS Corporate Branding Guidelines

The U.S. Postal Service requests that it is referenced and acknowledged as the source of information for all U.S. Postal Service data that has been acquired through the Internet and/or from other sources. However, this is not mandatory. The following guidelines should be followed for those that want to authenticate and/or validate the data displayed from the U.S. Postal Service.

Preferred Reference

Use one of the following when the USPS is the only referenced source:

- "Information provided by www.usps.com"

  or

- Use the official USPS corporate logo or USPS product-specific logos.

Digital copies of USPS corporate trademarks/logos are available through the U.S. Postal Service, Public Policy and Communications Department, Washington, D.C. You can request the USPS corporate logo and/or product-specific logos by e-mailing ilogo@email.usps.gov. Requests will be responded to by e-mail within 10 days. We will review your web site, and if appropriate, provide the logo for usage in accordance with these guidelines and the license grant contained in the Terms and Conditions of Use for Internet Shipping Application Program Interfaces (APIs). If your web page is not available over the Internet, please provide a screen shot of the page where the logo will reside.

When requesting logo(s) you must provide the following information:

- company name
- URL and page where logo will reside
- type of business
- how and where the logo will be used
- contact name
- telephone number
- e-mail address
- desired graphic format, e.g., GIF, TIF, JPEG, etc.
- logo desired:
  ____USPS Corporate Eagle Logo
  ____Express Mail
  ____Priority Mail
  ____other (describe)

Alternative Reference

Use one of the following when the USPS is listed with other shipping carriers or web sites:

- United States Postal Service
- U.S. Postal Service
- U.S.P.S. (use period after each initial)

The above alternatives are listed in the order of U.S. Postal Service preference.

Examples:

"U.S. Postal Service delivery standard is two days."

"U.S.P.S. Priority Mail rate is $3.95."

Registered Trademarks

The USPS trademarks listed on page i, as well as any logos requested from USPS Public Policy and Communications Department, should not be altered or abbreviated. You can request product-specific logos by e-mailing ilogo@email.usps.gov.

Trademark Ownership and Use

USPS trademarks are trademarks owned solely and exclusively by USPS and may be used only in the form, manner and with appropriate legends prescribed by USPS. All advertising and other uses of USPS trademarks must include a legend indicating that USPS trademarks are the property of USPS and that they are being used under license from USPS, together with any other legends or marking that may be required by law. Nothing contained in this document shall be deemed to convey any title or ownership interest to any user except for the nonexclusive rights granted under the Terms and Conditions of Use for Internet Shipping Application Program Interfaces and this document. All use of USPS Trademarks shall inure to the benefit of USPS.

Getting Started

Administrative Steps

For each step described below, it is indicated which of the APIs listed in the *Introduction* section requires the action(s) in the step. Many of the steps apply to all APIs.

Step 1: Register

Applicable APIs: All

To use the USPS APIs you must be a registered user. If you have not registered, go to http://www.usps.com/webtools/ and follow the instructions to register for the APIs.

Upon completion of the registration process, your user ID will be sent via e-mail to the address specified in the registration. You will be immediately granted access to the production server for our price calculators, package tracking, address information and service standards and commitments APIs. Please refer to the Restrictions on page 1 for an important notice regarding the use of your user ID.

Step 2: Obtain a Merchandise Return Service Permit

Applicable APIs:

Electronic Merchandise Return Service and
Electronic Merchandise Return Service with Delivery Confirmation

This step is only required if you are implementing the Electronic Merchandise Return (EMRS) Service API (with or without Delivery Confirmation). If you are implementing any other API, skip this step.

A Merchandise Return Service permit is required for Electronic Merchandise Return Service. Merchandise Return Service may be established at any post office in the United States and its territories and possessions, or at any U.S. military post office overseas (APO/FPO). It is not available for any foreign country.

If you do not have a valid Merchandise Return Service permit, you need to submit the following items to the local post office that will be delivering your return merchandise:

1. a completed USPS Form 3615
2. payment for the annual permit fee
3. The sample Electronic Merchandise Return Service Notification label provided in the *Administrative Forms* section.

Print the Notification label and submit to your local post office ***as is***. Do not alter the image.

Important: When printing PDF files with barcodes, be sure that the "Fit to Page" option in the print dialogue box of Adobe Acrobat Reader is unchecked.

Step 3: Address API Permissions

Applicable APIs: Address Verification, City State Lookup, Zip Code Lookup

The Web Tools Address APIs are intended to be used on a transactional basis—i.e. no database cleansing—in conjunction with USPS shipping. Upon agreement to these terms and conditions customers will be granted access automatically when registering for Web Tools.

Step 4: Run "Live" XML from Production Server

Applicable APIs: All

At this point, you have completed all required initiation steps and are now ready to send "live" data to the production server and begin full API service. The Web Tool Kit User's Guide for the API you are using has XML schema information.

Administrative Forms

The following forms are provided in this section:

1. **Licensing Agreement for the API Connector Code**

   Print the Licensing Agreement and e-mail the completed agreement to uspstechnicalsupport@mailps.custhelp.com.

2. **USPS API Printer Certification Submission Form**

   Print and complete the Printer Certification Submission form and mail, along with the ten barcoded labels, to:

   National Customer Support Center
   Attn: Barcode Certification - Web Tools (API)
   United States Postal Service
   225 N Humphreys Blvd Ste 501
   Memphis, TN 38188-1001

   You can either print the form from this PDF file and fill out with a pen, or copy the form to a Microsoft Word file and fill out electronically before printing and signing.

3. **Electronic Merchandise Return Service Notification Label**

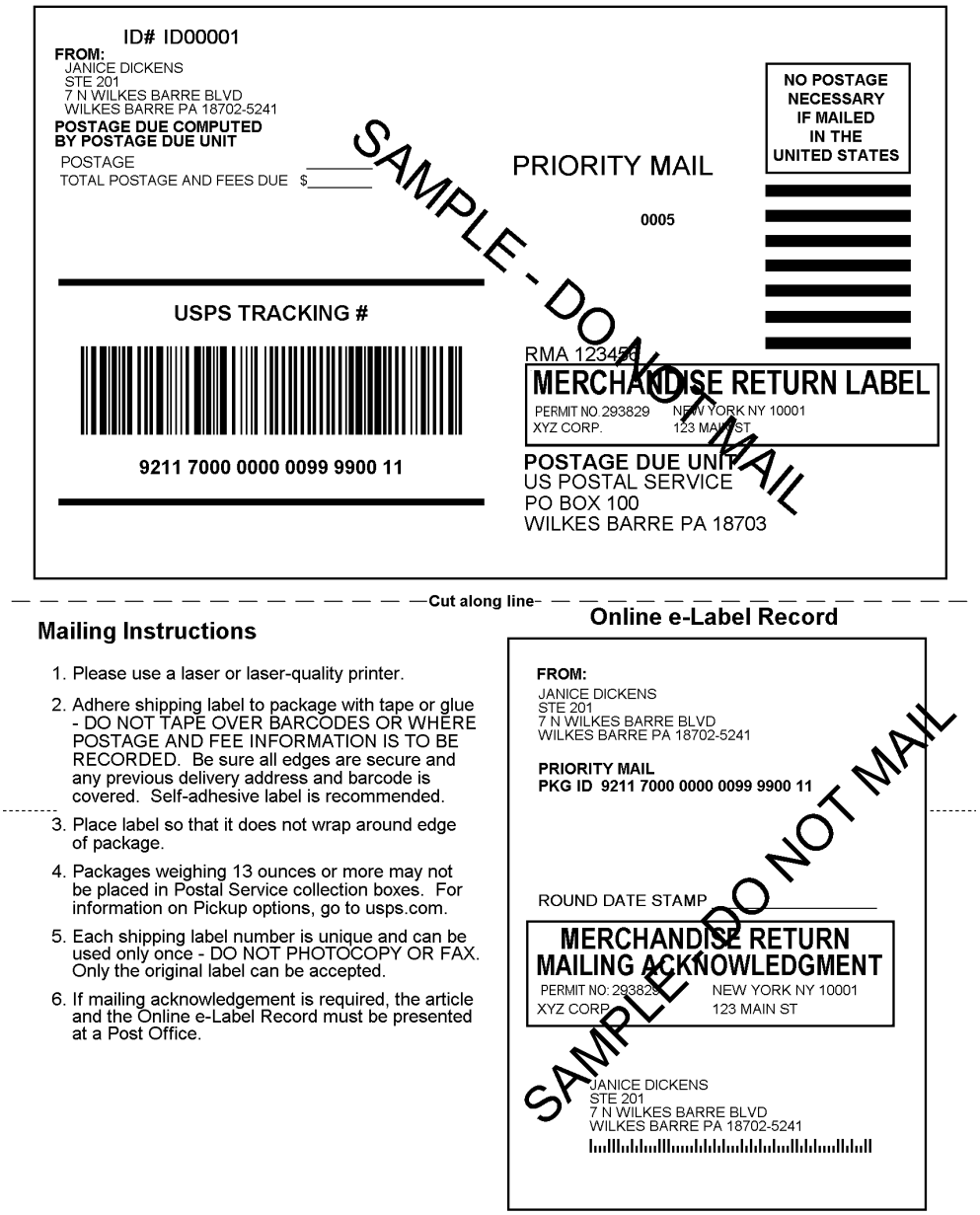
   Print the Notification label and submit to your local post office ***as is***. Do not alter the image. There is nothing to fill out or complete.

Important: When printing PDF files with barcodes, be sure that the "Fit to Page" option in the print dialogue box of Adobe Acrobat Reader is unchecked.

Electronic Merchandise Return Service Notification

Attention: Postmaster/Mailing Requirements

The holder of this sample Merchandise Return Service label will be using the USPS Internet Shipping Application Program Interface (API) program to prepare and generate Merchandise Return Service labels. Please provide a Merchandise Return Service Permit as per DMM 507.11.0. If you need additional information regarding this program, contact the USPS Internet Customer Care Center at 1-800-344-7779.



(The above image is not to scale.)

Technical Instructions

Client Protocols

HTTP/HTTPS protocols are used for exchange of data. This is to facility passage through corporate firewalls and inclusion of multiple technologies for implementation. The simple example of this is a URL in a standard browser of the form:

```
http://production.shippingapis.com/ShippingAPI.dll?
API=APINAME&XML=<APINAMEREQUEST USERNAME='your account'>
<tag>data</tag><tag1>data</tag1></APINAMEREQUEST>
```

At the other extreme, coding to utilize TCP/IP sockets will also provide the basis for a solution though it should be considered only for extreme processing conditions.

In the middle of the technology curves are the Java objects and Microsoft ActiveX objects that are documented and robust enough for production use. For java, please reference the Apache Jakarta Project at http://jakarta.apache.org/. For Microsoft, reference material can be located on MSDN at http://msdn.microsoft.com/. Search for Windows HTTP Services to locate the WinHTTP reference.

Certain APIs have Personally Identifiable Information (PII) that requires the use of secure HTTP connections under USPS policies. Any API that has a TO: /FROM: address with package information is classified as a PII policy API. Requests for these APIs are sent to a secure server group using a URL of the form:

```
https://secure.shippingaps.com/ShippingAPI.dll?
API=APINAME&XML=<APINAMERequest USERNAME='your account'>
<tag>data here</tag><tag1>data</tag1></APINAMERequest>
```

XML Overview

The exchanges presented in this document are presented in XML, since that is how the actual data will be structured and transmitted, and it is also a convenient method to document this interface. XML uses a hierarchical (tree) element structure. Each element consists of a start tag of the form `<Name>`, and an end tag of the form `</Name>`, between which can be data and other elements. `<Name/>` is shorthand for `<Name></Name>`, an element with no data. Attributes such as USERID can be included in the start tag. *All data and attribute values in this document are for illustration purposes and are to be replaced by the actual values.* Developers must use the order and case for tag names of the sample code contained in this document. However, the tabs and carriage returns in the XML structures are for readability only; there is no need for white space in the actual transmissions.

The XML API offers an interface that enables both request(s) and responses to be fully structured. As shown in the following example, XML's set of self-defining tags allows multiple packages to be tracked with a single request. The ID field is used to match a particular entity in the request with the corresponding entity in the response.

For the latest information on XML from Microsoft and other leading vendors, browse:

- http://www.w3.org/XML
- http://www.xml.com

Software Development

An API request is the start of a transaction that concludes when the response is returned. Every request is stateless and no cookies or URL rewrites are used during the exchange. The easiest model to describe is that a request behaves like an HTML page with a <FORM> submission. An example of this looks as such:

```
<HTML>
<HEAD></HEAD>
<BODY>
<FORM method="POST" NAME="MYFORM"
ACTION="http://uspsshippinagonlineserver/receiver.dll">
<INPUT TYPE="TEXT" NAME="API" SIZE="25">
<INPUT TYPE="TEXT" NAME="XML" SIZE="8000">
<INPUT TYPE="SUBMIT" NAME="GO" VALUE="GO">
</FORM>
</BODY>
</HTML>
```

The web servers will accept either ***GET or POST http requests.

The return content is XML structured data Refer to the Apache XML web site at http://xml.apache.org/ for further information on character encodings and characteristics. A return will have this form (formatted for readability):

```
<?XML version="1.0" ?>
<APINAMEResponse>
<tag>return data</tag><tag1>additional</tag1>
</APINAMEResponse>
```

All request and response tags are case sensitive and misspelled tags will generate errors. In addition, order is important because of the coordination of some requests require specific tags in a specific order.

Label requests return the label in the requested format encoded as base64 text within the XML. Microsoft XML Core Services can be used to decode the text stream into a binary representation of the requested image. Currently decoding will generate a TIFF (Tagged Information File Format with CCITT group4 compression) or a PDF (http://www.adobe.com/).

**** Note**: GET http requests have length restrictions, whereas POST http requests do not. Please take this under consideration when determining the request-response method that you choose.*

Error Handling

When an error condition exists, a specific XML return is generated. The following example shows the tags that are returned:

```
<?xml version="1.0"?>
<Error>
  <Number>-2147217951</Number>
  <Source>EMI_Respond :EMI:clsEMI.ValidateParameters:
       clsEMI.ProcessRequest;SOLServerIntl.EMI_Respond</Source>
  <Description>Missing value for To Phone number.</Description>
  <HelpFile></HelpFile>
  <HelpContext>1000440</HelpContext>
</Error>
```

For APIs that can handle multiple transactions, the error conditions for requests for multiple responses to be returned together are handled at the response level. For example: a request for rates for two packages. If the addresses are non-existent, an "Error document" is returned to the user. On the other hand, if the address for the first package is acceptable but not the second, the response document contains the information for the first address, but under the XML tag for the second address there is an error tag.

Errors that are further down in the hierarchy also follow the above format.

There is not a compendium of error messages or states that is available. The <Description> tag has the identified problem and should be used to diagnosis the situation. Problems can arise even after thoroughly testing an application because of changes to USPS rates, policies or procedures may be implemented periodically may generate errors that didn't occur previously. **If a request to the ICCC is made for help with diagnosing a problem please have both the response XML and the request XML available.**

Testing

There is no capacity for testing in the USPS Web Tools infrastructure. Any account performing capacity/stress testing may be terminated.

A minimal testing function is available for client applications. Some APIs have canned tests that use an explicit XML request and allow for the testing of general programming techniques. As these tests are explicit, they are identified as available only for selected APIs. Processing returns, errors, and parsing techniques can be programmatically done using these canned test scenarios,

There are categories of APIs that provide repeatable responses such as the domestic and international rates APIs or address verification. These APIs are the easiest because of the small number of tags. These APIs are included in the default permissions.

APIs that return labels or barcode information that must be printed become available after applying for advanced permissions from the ICCC. To test label accuracy, a label certification API is available for each label type. The certify APIs can also be used for limited testing because "live" labels are not generated.

Starter APIs

The URLs in the following tests represent a sequence that is a canned test and the response is essentially hard coded to provide a specific return data set. Four API sets are selected to familiarize the programmer with expected processing from an API request. Address Info set is the most used API group because of address validation. Track is the second most useful API because of package tracking. The third most useful set is Express Mail Commitments. This API returns a multiple group response and is one of the more complex returns because of the variations of the content. The last set is for Package Pickup services. This API cannot be tested in the production environment. This API group is used to schedule pickup requests that are routed to a carrier for processing. Incorrect use of the API will cause a carrier to attempt to pick up a parcel from the address in the request.

The code example located at the end of the document can be cut and pasted into Notepad and saved as an HTML file. Through the Internet Explorer browser, the tests can be cut and pasted into the HTML page and actually submitted. Replace the USERID value with the information received from the registration process.

Address Info

Test Request #1

This test "cleanses" an address and provides the ZIP+4 value.

```
http://production.shippingapis.com/ShippingAPITest.dll?API=Verify
&XML=<AddressValidateRequest USERID="xxxxxxx"><Address
ID="0"><Address1></Address1>
<Address2>6406 Ivy Lane</Address2><City>Greenbelt</City><State>MD</State>
<Zip5></Zip5><Zip4></Zip4></Address></AddressValidateRequest>
```

```
<?xml version="1.0"?>
<AddressValidateResponse><Address ID="0"><Address2>6406 IVY
LN</Address2><City>GREENBELT</City><State>MD</State><Zip5>20770</Zip5>
<Zip4>1440</Zip4></Address></AddressValidateResponse>
```

Test Request #2

This test will also "cleanse" the address and completes the ZIP Code

```
http://production.shippingapis.com/ShippingAPITest.dll?API=Verify
&XML=<AddressValidateRequest USERID="xxxxxxx"><Address ID="1"><Address1>
</Address1><Address2>8 Wildwood Drive</Address2><City>Old Lyme</City>
<State>CT</State><Zip5>06371</Zip5><Zip4></Zip4></Address>
</AddressValidateRequest>
```

```
<?xml version="1.0"?>
<AddressValidateResponse><Address ID="1"><Address2>8 WILDWOOD DR</Address2>
<City>OLD LYME</City><State>CT</State>
<Zip5>06371</Zip5><Zip4>1844</Zip4></Address></AddressValidateResponse>
```

Test Request #3

This API is used to find the City and State associated with a ZIP Code.

```
http://production.shippingapis.com/ShippingAPITest.dll?API=CityStateLookup
&XML=<CityStateLookupRequest USERID="xxxxxxx"><ZipCode ID= "0">
<Zip5>90210</Zip5></ZipCode></CityStateLookupRequest>

<?xml version="1.0"?>
<CityStateLookupResponse><ZipCode ID="0"><Zip5>90210</Zip5>
<City>BEVERLY HILLS</City><State>CA</State></ZipCode>
</CityStateLookupResponse>
```

Test Request #4

This test demonstrates the use of characteristic identifiers to allow grouping multiple requests into the same transaction.

```
http://production.shippingapis.com/ShippingAPITest.dll?API=CityStateLookup
&XML=<CityStateLookupRequest USERID="xxxxxxx"><ZipCode ID="5">
<Zip5>20770</Zip5></ZipCode></CityStateLookupRequest>

<?xml version="1.0"?>
<CityStateLookupResponse><ZipCode ID="5"><Zip5>20770</Zip5>
<City>GREENBELT</City><State>MD</State></ZipCode></CityStateLookupResponse>
```

Test Request #5

```
http://production.shippingapis.com/ShippingAPITest.dll?API=ZipCodeLookup
&XML=<ZipCodeLookupRequest USERID="xxxxxxx"><Address ID="0">
<Address1></Address1><Address2>6406 Ivy Lane</Address2>
<City>Greenbelt</City><State>MD</State></Address></ZipCodeLookupRequest>

<?xml version="1.0"?>
<ZipCodeLookupResponse><Address ID="0"><Address2>6406 IVY LN</Address2>
<City>GREENBELT</City><State>MD</State><Zip5>20770</Zip5><Zip4>1440</Zip4>
</Address></ZipCodeLookupResponse>
```

Test Request #6

```
http://production.shippingapis.com/ShippingAPITest.dll?API=ZipCodeLookup
&XML=<ZipCodeLookupRequest USERID="xxxxxxx">
<Address ID="1"><Address1></Address1>
<Address2>8 Wildwood Drive</Address2><City>Old Lyme</City><State>CT</State>
</Address></ZipCodeLookupRequest>

<?xml version="1.0"?>
<ZipCodeLookupResponse><Address ID="1"><Address2>8 WILDWOOD DR</Address2>
<City>OLD LYME</City><State>CT</State>
<Zip5>06371</Zip5><Zip4>1844</Zip4></Address></ZipCodeLookupResponse>
```

Track

Test Request #1

This test shows a multi-entry return that is arranged in reverse chronological order. Note that a DOM parser may scramble the order of the XML which may cause programmatic confusion.

```
http://production.shippingapis.com/ShippingAPITest.dll?API=TrackV2
&XML=<TrackRequest USERID="xxxxxxxx">
<TrackID ID="EJ958083578US"></TrackID></TrackRequest>
```

```
<?xml version="1.0"?>
<TrackResponse><TrackInfo ID="EJ958083578US"><TrackSummary>
Your item was delivered at 8:10 am on June 1 in Wilmington DE 19801.
</TrackSummary><TrackDetail>
May 30 11:07 am NOTICE LEFT WILMINGTON DE 19801.
</TrackDetail><TrackDetail>
May 30 10:08 am ARRIVAL AT UNIT WILMINGTON DE 19850.
</TrackDetail><TrackDetail>
May 29 9:55 am ACCEPT OR PICKUP EDGEWATER NJ 07020.
</TrackDetail></TrackInfo></TrackResponse>
```

Test Request #2

```
http://production.shippingapis.com/ShippingAPITest.dll?API=TrackV2
&XML=<TrackRequest USERID="xxxxxxxx">
<TrackID ID="EJ958088694US"></TrackID></TrackRequest>
```

```
<?xml version="1.0"?>
<TrackResponse><TrackInfo ID="EJ958088694US"><TrackSummary>
Your item was delivered at 1:39 pm on June 1 in WOBURN MA 01815.
</TrackSummary><TrackDetail>
May 30 7:44 am NOTICE LEFT WOBURN MA 01815.
</TrackDetail><TrackDetail>
May 30 7:36 am ARRIVAL AT UNIT NORTH READING MA 01889.
</TrackDetail><TrackDetail>
May 29 6:00 pm ACCEPT OR PICKUP PORTSMOUTH NH 03801.
</TrackDetail></TrackInfo></TrackResponse>
```

Express Mail Commitment

Test Request #1

The API return is an example of a complex return where there are nested tags.

```
http://production.shippingapis.com/ShippingAPITest.dll?
API=ExpressMailCommitment&XML=
<ExpressMailCommitmentRequest USERID="xxxxx">
      <OriginZIP>207</OriginZIP>
      <DestinationZIP>11210</DestinationZIP>
      <Date></Date>
</ExpressMailCommitmentRequest>
```

```
<?xml version="1.0"?>
<ExpressMailCommitmentResponse><OriginZIP>207</OriginZIP>
<OriginCity>GREENBELT</OriginCity><OriginState>MD</OriginState>
```

```
<DestinationZIP>11210</DestinationZIP>
<DestinationCity>BROOKLYN</DestinationCity>
<DestinationState>NY</DestinationState><Date>05-Aug-2004</Date>
<Time>11:30 AM</Time><Commitment>
<CommitmentName>Next Day</CommitmentName>
<CommitmentTime>3:00 PM</CommitmentTime>
<CommitmentSequence>A0115</CommitmentSequence>
<Location><CutOff>6:00 PM</CutOff>
<Facility>EXPRESS MAIL COLLECTION BOX</Facility>
<Street>119 CENTER WAY</Street><City>GREENBELT</City>
<State>MD</State><Zip>20770</Zip></Location>
<Location><CutOff>3:00 PM</CutOff>
<Facility>EXPRESS MAIL COLLECTION BOX</Facility>
<Street>7500 GREENWAY CENTER DRIVE</Street><City>GREENBELT</City>
<State>MD</State><Zip>20770</Zip></Location>
</Commitment></ExpressMailCommitmentResponse>
```

Test Request #2

The test is similar to the one above except the return has multiple nested tags that represent additional commitments that are available based upon one of the parameters in the request.

```
http://production.shippingapis.com/ShippingAPITest.dll?
API=ExpressMailCommitment&XML=
<ExpressMailCommitmentRequest USERID="xxxxx"><OriginZIP>20770</OriginZIP>
<DestinationZIP>11210</DestinationZIP><Date>05-Aug-2004</Date>
</ExpressMailCommitmentRequest>

<?xml version="1.0"?>
<ExpressMailCommitmentResponse><OriginZIP>20770</OriginZIP>
<OriginCity>GREENBELT</OriginCity><OriginState>MD</OriginState>
<DestinationZIP>11210</DestinationZIP>
<DestinationCity>BROOKLYN</DestinationCity>
<DestinationState>NY</DestinationState><Date>05-Aug-2004</Date>
<Time>11:30 AM</Time><Commitment>
<CommitmentName>Next Day</CommitmentName>
<CommitmentTime>3:00 PM</CommitmentTime>
<CommitmentSequence>A0115</CommitmentSequence><Location>
<CutOff>6:00 PM</CutOff>
<Facility>EXPRESS MAIL COLLECTION BOX</Facility>
<Street>119 CENTER WAY</Street><City>GREENBELT</City>
<State>MD</State><Zip>20770</Zip></Location>
<Location><CutOff>3:00 PM</CutOff>
<Facility>EXPRESS MAIL COLLECTION BOX</Facility>
<Street>7500 GREENWAY CENTER DRIVE</Street><City>GREENBELT</City>
<State>MD</State><Zip>20770</Zip></Location></Commitment>
<Commitment><CommitmentName>Next Day</CommitmentName>
<CommitmentTime>12:00 PM</CommitmentTime>
<CommitmentSequence>A0112</CommitmentSequence><Location>
<CutOff>6:00 PM</CutOff>
<Facility>EXPRESS MAIL COLLECTION BOX</Facility>
<Street>119 CENTER WAY</Street><City>GREENBELT</City>
<State>MD</State><Zip>20770</Zip></Location><Location>
<CutOff>3:00 PM</CutOff>
<Facility>EXPRESS MAIL COLLECTION BOX</Facility>
```

```
<Street>7500 GREENWAY CENTER DRIVE</Street><City>GREENBELT</City>
<State>MD</State><Zip>20770</Zip></Location><Location>
<CutOff>9:45 PM</CutOff><Facility>AIR MAIL FACILITY</Facility>
<Street>ROUTE 170 BLDG C DOOR 19</Street><City>BALTIMORE</City>
<State>MD</State><Zip>21240</Zip></Location></Commitment>
</ExpressMailCommitmentResponse>
```

Package Pickup

Test Request #1

```
https://secure.shippingapis.com/ShippingAPITest.dll?
API=CarrierPickupAvailability&XML=
<CarrierPickupAvailabilityRequest USERID="XXXX">
<FirmName>ABC Corp.</FirmName>
<SuiteOrApt>Suite 777</SuiteOrApt>
<Address2>1390 Market Street</Address2>
<Urbanization></Urbanization>
<City>Houston</City>
<State>TX</State>
<ZIP5>77058</ZIP5>
<ZIP4>1234</ZIP4>
</CarrierPickupAvailabilityRequest>
```

```
<?xml version="1.0"?>
<CarrierPickupAvailabilityResponse><FirmName>ABC Corp.</FirmName>
<SuiteOrApt>Suite 777</SuiteOrApt>
<Address2>1390 Market Street</Address2>
<Urbanization></Urbanization><City>HOUSTON</City><State>TX</State>
<ZIP5>77058</ZIP5><ZIP4>1234</ZIP4><DayOfWeek>Monday</DayOfWeek>
<Date>3/1/2004</Date><CarrierRoute>C</CarrierRoute>
</CarrierPickupAvailabilityResponse>
```

Test Request #2

```
https://secure.shippingapis.com/ShippingAPITest.dll?
API=CarrierPickupAvailability&XML=
<CarrierPickupAvailabilityRequest USERID="XXXX">
<FirmName></FirmName>
<SuiteOrApt></SuiteOrApt>
<Address2>1390 Market Street</Address2>
<Urbanization></Urbanization>
<City></City>
<State></State>
<ZIP5>77058</ZIP5>
<ZIP4></ZIP4>
</CarrierPickupAvailabilityRequest>
```

```
<?xml version="1.0"?>
<CarrierPickupAvailabilityResponse><FirmName></FirmName>
<SuiteOrApt></SuiteOrApt><Address2>1390 Market Street</Address2>
<Urbanization></Urbanization><City>HOUSTON</City><State>TX</State>
<ZIP5>77058</ZIP5><ZIP4>1234</ZIP4><DayOfWeek>Monday</DayOfWeek>
<Date>3/1/2004</Date><CarrierRoute>C</CarrierRoute>
</CarrierPickupAvailabilityResponse>
```

Testing Tips

- Check for proxy servers that may be blocking access.
- Some APIs require HTTPS protocol because of PII. If SSL connections are failing, check that the software is accepting the certificate.
- Most APIs complete in a few seconds but some APIs require the compilation of a lot of interrelated data and may require significantly more time. Use a network trace utility to determine transport time and provide a cushion for timeouts.
- Use a browser to verify request/response interaction if the client application doesn't provide some form of debug tracking.
- ISO-8859-1 encoding is the expected character set for the request. Make sure that special characters embedded in any tag data such as & < > are escaped and url-encoded.
- Use the Domestic Mail Manual (DMM) and International Mail Manual (IMM) as a reference for package dimensions, weights and restrictions for the many types of service. The Postal Explorer web site at http://pe.usps.gov/ has online versions of these manuals.
- Keep registration contact information accurate. E-mail notices involving major updates are sent using the e-mail address and person listed as contacts.
- There are no sanctioned online forums, newsgroups or blogs for Web Tools APIs. The ICCC has current information as to the operational status of the Web Tools API service and can answer questions about registration and operation status and receive problem reports.

Updates to APIs

Periodically, the USPS Web Tools program has major updates based upon Postal Rate Case changes. This update may necessitate the introduction of new APIs, discontinuance of APIs due to mail service changes, and addition of optional tags to existing APIs. Notice will be sent via e-mail to the registered e-mail address for all users.

Address information for verification is updated weekly. Postal rates, restrictions, prohibitions and other information that effect international mailings are updated as soon as they become available in the IMM and DMM.

References

http://www.w3.org/XML
http://www.xml.com/
XML reference for use and SDK type tools for parsing.

http://pe.usps.gov/

US Postal Service reference for mailing services

http://www.usps.com/webtools/

Web Tools information pages that contain documents for the APIs, links to registration and other resources.

http://jakarta.apache.org/

Open Source reference for all types of technologies relevant to the use of Web Tools API.

CODE Example

The following code example is done as an HTML page that works under Microsoft's Internet Explorer. Copy the page to the clipboard and then paste into notepad or a text editor and save to a file. The browser issues warnings about ActiveX Objects and may require some adjustments to get the page to execute. The javascript code assumes that there are no firewalls or proxy servers between the client and the destination server that must be used to complete the communication path.

```
<html><head>
<title>Example courtesy of United States Postal Service</title>
</head><body>
<script type="text/javascript">
var objSrvHTTP = null;
var Version;
var HTTPVersions = new Array( "7.0","6.0","5.0","4.0","3.0" );
for( Version = 0; Version < HTTPVersions.length; Version++ ) {
  try {
    objSrvHTTP = new ActiveXObject("Msxml2.ServerXMLHTTP." +
HTTPVersions[Version]);
      window.status = "Using ServerXMLHTTP." + HTTPVersions[Version];
      break;
  } catch(e) {}
      objSrvHTTP = null;
}
if( objSrvHTTP == null ) { alert( "No HTTP object available" );}
else {
      objSrvHTTP.setOption( 2, 13056 );  // ignore certificate errors
      objSrvHTTP.setProxy(1);  // ignore proxy servers (see proxycfg utility)
      objSrvHTTP.setTimeouts( 10000, 10000, 10000, 10000 ); //10 sec timeout
}
function getAPIResponse( style ) {
    var URL = (style) ? APIForm.XMLContent.value :

((APIForm.secure.checked)?"https://":"http://")+APIForm.APIServer.value+
       APIForm.tURI.value+"?API="+APIForm.APIName.value+"&XML="+
       APIForm.XMLContent.value;
// alert( URL );
      try {objSrvHTTP.open ("GET", URL, false); objSrvHTTP.send ();
            XMLResponse.innerText = (objSrvHTTP.status == 200) ?
                  objSrvHTTP.responseText : "HTTP Error " +
objSrvHTTP.status;}
      catch(e) { XMLResponse.innerText = "Error condition " + e.description;
          objSrvHTTP = new
ActiveXObject("Msxml2.ServerXMLHTTP."+HTTPVersions[Version]);}
}
</script>
<form name="APIForm" method="get">
<label for="APIName">API Name</label>
<input type="text" name="APIName" size="25" style="margin:0 51px 0 10px">
<button onClick="getAPIResponse(false);">Build URL</button><br>
<label for="APIServer">API Server</label>
<input type="text" size="30" name="APIServer" style="margin:0 13px 0 7px;">
<input name="secure" type="checkbox">
<label for="secure"> use https</label><br>
<label for="tURI">URI path</label>
<input name="tURI" type="text" value="/ShippingAPITest.dll" style="margin:0
21px 0 20px;" size="30">
<button onClick="getAPIResponse(true);">Text is URL</button><br>
<textarea id="XMLContent" name="XMLContent" cols="80" rows="5"></textarea>
</form>
<hr>
<div style="width:100%"><span id="XMLResponse"></span></div>
</body></html>
```