



United States Postal Service® Web Tool Kit User's Guide



A Technical Guide to

Signature Confirmation Label

&

Signature Confirmation Barcode Number

Application Programming Interfaces



Before implementing either of these APIs, the *Administrative Guide for Application Programming Interfaces* must be read.

Version 1.3 (8/5/02)

To Our Customers

In the e-mail that accompanied this guide you received a password and user ID that will allow you to begin sending calls to the “test server” when you are ready. 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 manned from 7:00AM to 11:00PM EST.

E-mail: icustomer@usps.com

Telephone: 1-800-344-7779 (7:00AM to 11:00PM EST)

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 7 x 24 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

Trademarks

Express Mail, First-Class Mail, Global Priority Mail, Parcel Post, Parcel Select, Priority Mail, USPS, and ZIP + 4 are registered trademarks of the U.S. Postal Service.

Global Express Guaranteed, Global Express Mail, GXG, International Parcel Post, Priority Mail Global Guaranteed, Signature Confirmation, and ZIP Code are trademarks of the U.S. Postal Service.

Microsoft, Visual Basic, and Word are registered trademarks of Microsoft Corporation.

Adobe Acrobat is a trademark of Adobe Systems Incorporated.

©Copyright 2002 United States Postal Service

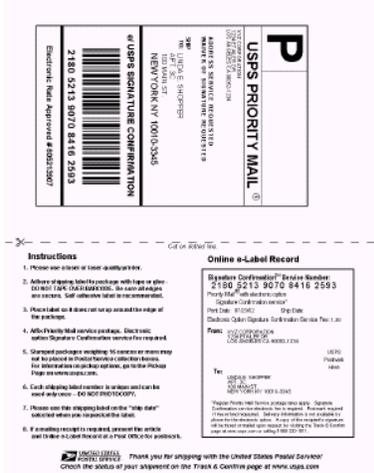
Table of Contents

Introduction to the Signature Confirmation Label API	1
Label Options.....	3
Pay At Delivery Service.....	6
User ID and Password Restrictions	6
Transaction Procedures	7
Technical Steps	7
Step 1: Build the XML Request	7
"Cannd" Test Requests	7
Valid Test Requests	8
Pre-Defined Error Requests	10
XML Tags and Values Allowed (Required & Optional)	12
"Sample" Request	15
"Live" Request.....	17
Visual Basic Request.....	18
Steps 2 & 3: Make the Internet Connection and Send the XML Request.....	22
Using HTTP Connection DLL	22
Using WinInet.....	22
Step 4: Unpack the XML Response	24
Types of Responses	24
Using Visual Basic	24
Base64 Decoding in VBScript in an ASP	25
Errors	26
Output	27
"Cannd" Test Responses	27
"Sample" Responses.....	30
"Live" Responses	31
XML Output Example	31
Mailing Instructions and Label Placement on Packages.....	32
Introduction to the Signature Confirmation Barcode Number API.....	33
User ID and Password Restrictions	33
Transaction Procedures	33
Technical Steps	34
Step 1: Build the XML Request	34
XML Tags and Values Allowed (Required & Optional)	34
"Sample" Request	36
"Live" Request.....	37
Steps 2 & 3: Make the Internet Connection and Send the XML Request.....	38
Using HTTP Connection DLL	39
Using WinInet.....	39
Step 4: Unpack the XML Response	40
Types of Responses	40

Using Visual Basic	40
Errors	41
Output	42
Responses	42
"Sample" XML Response	42
"Live" XML Response	42
Generating Your Own Label	43
USPS Barcode Requirements	44
Text	44
Numbers	44
Identification Bars	44
Human-Readable Description of Mailpiece Specifications	44
Customer Online Record Requirements	45
Mailing Instructions and Label Placement on Packages	45
Appendix: Signature Confirmation Product Tracking System Extract File Retrieval	47

Introduction to the Signature Confirmation Label API

With the USPS's Signature Confirmation, you (or your customers) can access information on the Internet about the delivery status of First-Class Mail parcels, Priority Mail and Package Services (Parcel Post, Bound Printed Matter, Media Mail, and Library Mail), including the date, time, and ZIP Code of delivery, as well as attempted deliveries, forwarding, and returns. Signature Confirmation service is not available to APO/FPO addresses, foreign countries, or most U.S. territories.



There are two separate APIs that generate a Signature Confirmation number:

1. The Signature Confirmation Label API generates a label (and accompanying Customer Online Record), either with or without shipping addresses and information depending on the request. The label returned by the API is printed by the sender and attached to the package.
2. The Signature Confirmation Barcode Number API generates a Signature Confirmation number that can be applied to user-generated labels. This API only returns the number, which must be used when exercising the option of generating a custom label.

The charge (known as the “electronic option rate”) for Signature Confirmation is \$1.30 for Priority Mail, First-Class mail parcels, and Package Services parcels.

There are three ways to obtain the delivery status of Signature Confirmation labels (barcodes) generated by these APIs:

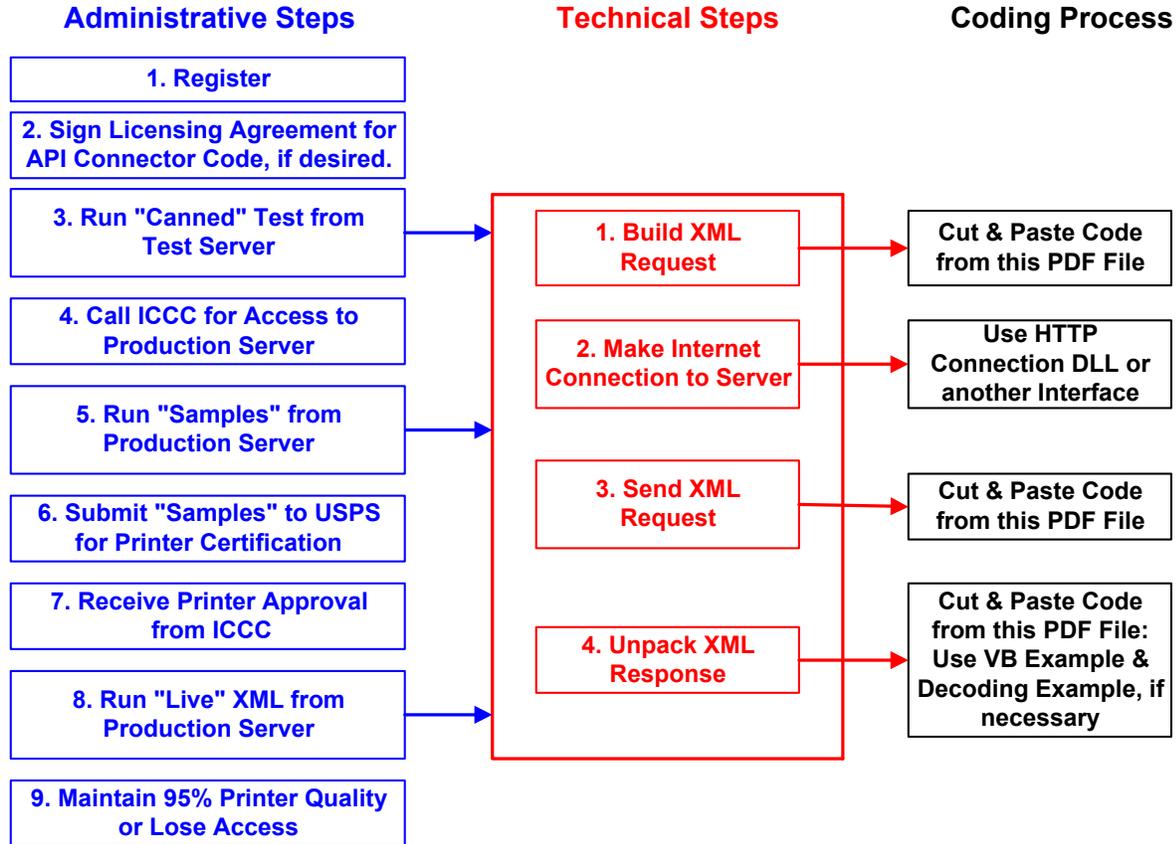
1. implementing the Track/Confirm API (available at <http://www.uspswebtools.com/>)
2. accessing the www.usps.com Track/Confirm pages
3. using the USPS Product Tracking System (see the Appendix, *Signature Confirmation PTS Tracking System Extract File Retrieval*)

In addition, you can obtain a copy of the recipient’s signature via mail or FAX by visiting the Track/Confirm pages at www.usps.com, or by calling 1-800-222-1811.

This section describes the Signature Confirmation Label API. There is a separate section in the document that describes the Signature Confirmation Barcode Number API. Whichever API you choose, you must test against our test server and have your barcode label printer certified by the USPS before use. Refer to Administrative Step 5 in the *Administrative Guide for APIs*.

As shown in the illustration below, implementing either of the Signature Confirmation APIs requires a series of *Administrative Steps*. The *Administrative Guide for APIs*, also available at <http://www.uspswebtools.com/>, provides necessary information and procedures prior to installation. The illustration also shows the *Technical Steps* required to run XML transactions for the API to either the test server or the production server, as well as the *Coding Process* to be

followed for each *Technical Step*. This document provides step-by-step instructions for both the Technical Steps and Coding Process illustrated below. As each step is presented throughout this guide, appropriate portions of the illustration below will be repeated as a reference point in the implementation process.



Implementing these APIs requires experienced programmers who are familiar with Internet and web site development tools and techniques. Before implementing either of these APIs, the Administrative Guide for Application Programming Interfaces must be read.

Label Options

You have two label options when requesting Signature Confirmation:

Label Option 3: A graphic image is returned that will include a barcode, Package ID Code (PIC), return name and address, and delivery name and address (sample on the left, below).

Label Option 4: A graphic image is returned that will include a barcode and PIC number *without* the return and delivery name and address (sample on the right, below).



(The images above are not to scale.)

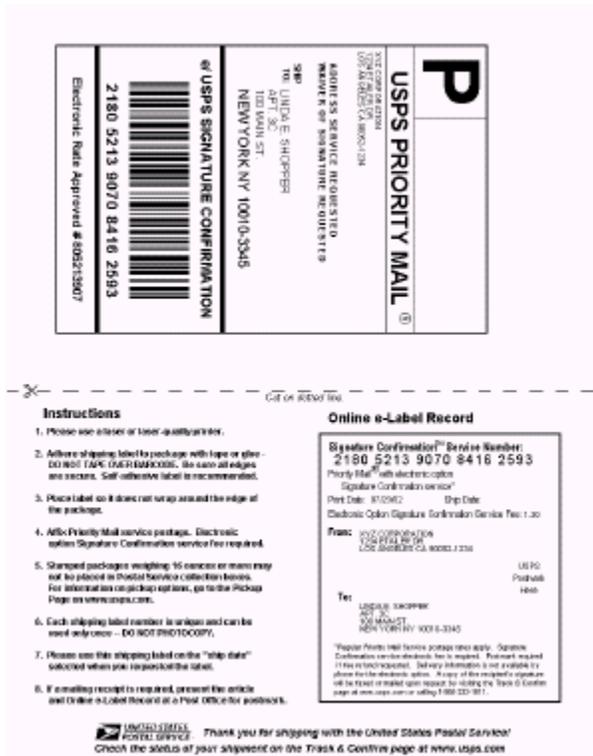
Both labels shown above contain a barcode and corresponding 20-digit Signature Confirmation number (known as the PIC). The components of the PIC number (2180 5213 9071 2582 9249 in the sample above right) are as follows:

Digits	Description	Comments
1-2	Service Type Code	Valid Service Type Codes are: 21 for Priority Mail and First-Class Mail parcels 22 for Package Services
3-11	API Program number	The API Program number is 805213907 (This number is used on every Signature Confirmation label).
12-19	Package ID #	The Package ID # is an 8-digit number generated by the Signature Confirmation API.
20	MOD 10 Check Digit #	The Check Digit is computed from the sequence number. For details on this check digit see Appendix G in the <i>Confirmation Services Technical Guide</i> , Publication 91, at http://www.usps.com/cpim/ftp/pubs/pub91.pdf or , for an HTML version, at http://www.usps.com/cpim/ftp/pubs/pub91/welcome.htm .

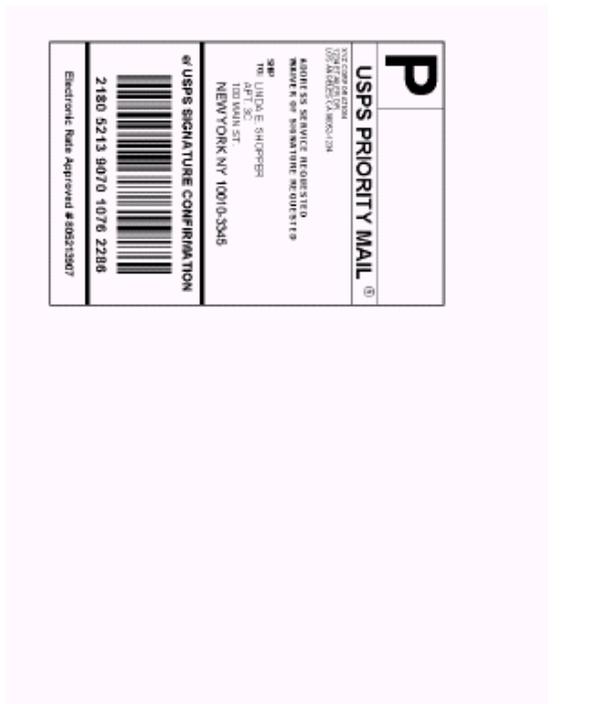
The PIC is used when requesting delivery status on a Priority Mail package or any of the Package Services. A daily manifest of all PICs issued is loaded by the USPS into the Product Tracking System so that customer inquiries on a package's delivery status made the day following the label date will receive a status response.

In addition to the Signature Confirmation shipping label, the API provides a printed Customer Online Record. There are two options in how the labels are returned—either together on the same page or separated and printed on two pages. Users also have the option of using the shipping information returned and generating their own labels using the Signature Confirmation Barcode Number API. See the *Introduction to the Signature Confirmation Barcode Number API* section.

For illustration purposes, there are two examples shown below. The top example shows Signature Confirmation labels returned on the same page. The bottom examples show Signature Confirmation labels returned on two pages. Note that the bottom examples show the labels split onto two separate pages but in the same horizontal configuration as printed in the combined example at the top of the page.



Example #1 (above): labels printed on same page. Example #2 (below): labels printed in same position, but on two separate pages.



(The above images are not to scale.)

Pay At Delivery Service

The USPS offers a service that uses USPS Confirmation Services and an escrow payment mechanism to effect a "Pay@Delivery Service." Pay@Delivery processing allows for the seller of an item and the buyer of that item to agree that a third-party will hold the buyer's funds until the USPS or the buyer has confirmed delivery of the package from the seller. When the USPS confirms delivery of the package via the USPS Confirmation Services, the money previously debited from the buyer's account will be credited to the seller's account. Those interested in licensing or utilizing this service should send an e-mail to <mailto:webtools@email.usps.com> describing their request. No USPS Web Tools APIs may be used, without prior approval, in a fashion mimicking the function of the USPS Pay@Delivery Service.

Warning - If users of USPS Web Tools APIs are shown to be violating the foregoing restriction on mimicking the function of the USPS Pay@Delivery Service, they will be subject to immediate loss of access to the USPS Web Tool server and termination of the licenses granted under the Terms and Conditions of Use.

User ID and Password Restrictions

The user ID and password that you have received are 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 and password are 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 and password, as well as this document and the APIs contained herein.

Warning: If the U.S. Postal Service discovers use of the same user ID and password 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's 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 password and user ID. Direct them to www.uspswebtools.com so that they can register, agree to the Terms and Conditions of Use agreement, and receive their own unique password and user ID.

Note to Software Distributors: The User ID and password 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 Distributors Policy Guide*, available at <http://www.uspswebtools.com/>.

For more information regarding the USPS Web Tool Kit password and user ID policy, or for questions regarding the distribution of documentation, send e-mail to icustomer@usps.com.

Transaction Procedures

The illustration below shows the transactional flow of information to and from the USPS Signature Confirmation Label API server.

Signature Confirmation Label API Server



INPUTS

(via XML Request from Customer to USPS)

From Name & Address
 Recipient Name & Address
 Service Type
 Label Option
 Package Weight
 Label Image Type
 Mailing Location (optional)
 Waiver of Signature (optional)
 Label Date (optional)
 Reference Number (optional)
 Address Service Request (optional)
 E-mail Information (optional)

SERVER TASKS

Builds XML Response
 Generates Label
 Encodes Label
 Tracks Packages
 Sends E-mail Notification

OUTPUTS

(via XML Response from USPS to Customer)

Confirmation Number
 Shipping Label
 Customer Online Record

Technical Steps

Step 1: Build the XML Request

“Canned” Test Requests

For testing purposes, the only values in the test code in this section that you should change are the “USERID,” “PASSWORD,” and “SERVERNAME.” Enter the user ID, password, and server name you received in the registration e-mail for *testing*. Your user ID and password never change, but the server name will change later when you send "sample" and “live” requests. The “live” server name will be provided when the ICCC provides you with access to the production server. *All remaining code in the test scripts provided below must remain unchanged.*

All of the test script code contained in this document can be cut and pasted for your use in testing the software. To copy the test script code from this PDF file, click on the icon for "Text Selector" and highlight the code. (The icon will look like



or



depending on your version of Adobe Acrobat.) You can then copy the code and paste it into your test document.

Valid Test Requests

There are four valid test requests below. Be sure to note the request numbers so you can match up the responses you will receive as provided in the "Canned" Test Responses section.

Test Request #1

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>3</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm>ABC Corp.</FromFirm>
  <FromAddress1>Ste 4</FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Greenbelt</FromCity>
  <FromState>MD</FromState>
  <FromZip5>20770</FromZip5>
  <FromZip4>4354</FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm>XYZ Corp.</ToFirm>
  <ToAddress1>Apt 303</ToAddress1>
  <ToAddress2>4411 Romlon Street</ToAddress2>
  <ToCity>Beltsville</ToCity>
  <ToState>MD</ToState>
  <ToZip5>20705</ToZip5>
  <ToZip4>5656</ToZip4>
  <WeightInOunces>22</WeightInOunces>
  <ServiceType>Parcel Post</ServiceType>
  <ImageType>TIF</ImageType>
</SignatureConfirmationRequest>
```

Test Request #2

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>4</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm></FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Greenbelt</FromCity>
  <FromState>MD</FromState>
  <FromZip5>20770</FromZip5>
  <FromZip4></FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm></ToFirm>
```

```
<ToAddress1></ToAddress1>
<ToAddress2>8 Wildwood Drive</ToAddress2>
<ToCity>Old Lyme</ToCity>
<ToState>CT</ToState>
<ToZip5>06371</ToZip5>
<ToZip4></ToZip4>
<WeightInOunces>22</WeightInOunces>
<ServiceType>Priority</ServiceType>
<ImageType>TIF</ImageType>
</SignatureConfirmationRequest>
```

Test Request #3

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>3</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm>ABC Corp.</FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Greenbelt</FromCity>
  <FromState>MD</FromState>
  <FromZip5>20770</FromZip5>
  <FromZip4></FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm></ToFirm>
  <ToAddress1></ToAddress1>
  <ToAddress2>8 Wildwood Drive</ToAddress2>
  <ToCity>Old Lyme</ToCity>
  <ToState>CT</ToState>
  <ToZip5>06371</ToZip5>
  <ToZip4>1844</ToZip4>
  <WeightInOunces>22</WeightInOunces>
  <ServiceType>Library Mail</ServiceType>
  <ImageType>PDF</ImageType>
</SignatureConfirmationRequest>
```

Test Request #4

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>4</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm></FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Greenbelt</FromCity>
  <FromState>MD</FromState>
  <FromZip5>20770</FromZip5>
  <FromZip4>3434</FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm>XYZ Corp.</ToFirm>
  <ToAddress1></ToAddress1>
  <ToAddress2>8 Wildwood Drive</ToAddress2>
  <ToCity>Old Lyme</ToCity>
  <ToState>CT</ToState>
  <ToZip5>06371</ToZip5>
  <ToZip4></ToZip4>
```

```
<WeightInOunces>22</WeightInOunces>
<ServiceType>Media Mail</ServiceType>
<ImageType>PDF</ImageType>
</SignatureConfirmationRequest>
```

Pre-Defined Error Requests

There are four pre-defined errors below. Be sure to note the request numbers so you can match up the responses you will receive as provided in the “*Canned*” Test Responses section.

Pre-defined Error Request #1: “Invalid Service Type”

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>3</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm></FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Brooklyn</FromCity>
  <FromState>NY</FromState>
  <FromZip5>11210</FromZip5>
  <FromZip4>3434</FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm>XYZ Corp.</ToFirm>
  <ToAddress1></ToAddress1>
  <ToAddress2>6406 Ivy Lane</ToAddress2>
  <ToCity>Greenbelt</ToCity>
  <ToState>MD</ToState>
  <ToZip5>20770</ToZip5>
  <ToZip4></ToZip4>
  <WeightInOunces>22</WeightInOunces>
  <ServiceType>Express</ServiceType>
  <ImageType>PDF</ImageType>
</SignatureConfirmationRequest>
```

Pre-defined Error Request #2: “Invalid Weight”

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>4</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm></FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Brooklyn</FromCity>
  <FromState>NY</FromState>
  <FromZip5>11210</FromZip5>
  <FromZip4>3434</FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm>XYZ Corp.</ToFirm>
  <ToAddress1></ToAddress1>
  <ToAddress2>6406 Ivy Lane</ToAddress2>
  <ToCity>Greenbelt</ToCity>
  <ToState>MD</ToState>
  <ToZip5>20770</ToZip5>
  <ToZip4></ToZip4>
```

```
<WeightInOunces>Twenty</WeightInOunces>
<ServiceType>Bound Printed Matter</ServiceType>
<ImageType>PDF</ImageType>
</SignatureConfirmationRequest>
```

Pre-defined Error Request #3: “Invalid Image Type”

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>3</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm></FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Brooklyn</FromCity>
  <FromState>NY</FromState>
  <FromZip5>11210</FromZip5>
  <FromZip4>3434</FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm>XYZ Corp.</ToFirm>
  <ToAddress1></ToAddress1>
  <ToAddress2>6406 Ivy Lane</ToAddress2>
  <ToCity>Greenbelt</ToCity>
  <ToState>MD</ToState>
  <ToZip5>20770</ToZip5>
  <ToZip4></ToZip4>
  <WeightInOunces>22</WeightInOunces>
  <ServiceType>Priority</ServiceType>
  <ImageType>ABC</ImageType>
</SignatureConfirmationRequest>
```

Pre-defined Error Request #4: “Label Date in Past”

```
<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>3</Option>
  <ImageParameters></ImageParameters>
  <FromName>John Smith</FromName>
  <FromFirm></FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Brooklyn</FromCity>
  <FromState>NY</FromState>
  <FromZip5>11210</FromZip5>
  <FromZip4>3434</FromZip4>
  <ToName>Jane Smith</ToName>
  <ToFirm>XYZ Corp.</ToFirm>
  <ToAddress1></ToAddress1>
  <ToAddress2>6406 Ivy Lane</ToAddress2>
  <ToCity>Greenbelt</ToCity>
  <ToState>MD</ToState>
  <ToZip5>20770</ToZip5>
  <ToZip4></ToZip4>
  <WeightInOunces>22</WeightInOunces>
  <ServiceType>Priority</ServiceType>
  <ImageType>TIF</ImageType>
  <LabelDate>02/28/2001</LabelDate>
</SignatureConfirmationRequest>
```

XML Tags and Values Allowed (Required & Optional)

The table below presents the *required* XML input tags for generating both “Sample” and “Live” requests and the restrictions on the values allowed. A second table is also included with *optional* tags for the Signature Confirmation API. An error message will be returned if an incorrect value is entered. Also, be aware of the maximum character amounts allowed for some tags. If the user enters more than those amounts, an error will not be generated. ***The API will simply pass in the characters up to the maximum amount allowed and disregard the rest.*** This is important since the resulting value could prevent delivery.

Required Signature Confirmation Tags

Input	XML Tag	Values Allowed
"Sample" Request	<SigConfirmCertifyRequest...	Input tag exactly as presented.
"Live" Request	<SignatureConfirmationRequest...	Input tag exactly as presented.
User ID	...USERID="userid"...	Use user ID provided with registration.
Password	...PASSWORD="password">	Use password provided with registration.
Label Option	<Option>	Enter one of the valid entries: "3" for Label Option #3 "4" for Label Option #4 See examples in <i>Label Options</i> section, above.
Printer Definition	<ImageParameters>	This tag is for future use. The tag is required, but there are no values to enter.
Name of Sender	<FromName>	Maximum characters allowed: 32
Company Name	<FromFirm>	This tag is required but the value is optional. Maximum characters allowed: 32
From Address Line 1	<FromAddress1>	Use this tag for an apartment or suite number. This tag is required but the value is optional. Maximum characters allowed: 32
From Address Line 2	<FromAddress2>	Use this tag for the primary address line. Maximum characters allowed: 32
From City	<FromCity>	Maximum characters allowed: 21
From State	<FromState>	Maximum characters allowed: 2
From ZIP Code	<FromZip5>	Input tag exactly as presented, not all caps. Maximum characters allowed: 5
From ZIP Code+4	<FromZip4>	Input tag exactly as presented, not all caps. This tag is required but the value is optional. Maximum characters allowed: 4
Name of Recipient	<ToName>	Maximum characters allowed: 38
Company Name	<ToFirm>	This tag is required but the value is optional. Maximum characters allowed: 38
To Address Line 1	<ToAddress1>	Use this tag for an apartment or suite number. This tag is required but the value is optional. Maximum characters allowed: 38
To Address Line 2	<ToAddress2>	Use this tag for the primary address line. Maximum characters allowed: 38
To City	<ToCity>	Maximum characters allowed: 21
To State	<ToState>	Maximum characters allowed: 2

To ZIP Code	<ToZip5>	Input tag exactly as presented, not all caps. Maximum characters allowed: 5
To ZIP Code+4	<ToZip4>	Input tag exactly as presented, not all caps. This tag is required but the value is optional. Maximum characters allowed: 4
Package Weight	<WeightinOunces>	Value must be numeric. Estimated weight is allowed.
Mail Service Type	<ServiceType>	Enter one of the valid entries: "Priority" (for Priority Mail) "First Class" (see below for definition) "Parcel Post" "Bound Printed Matter" "Media Mail" "Library Mail"
Label Image Type	<ImageType>	Enter one of the valid entries: "TIF" "JPG" "PDF" "GIF" "None" (if no label is desired; Signature Confirmation number must be used on custom labels)

First-Class Mail Parcels: For the purposes of adding Signature Confirmation service, a First-Class Mail parcel is defined as any piece that:

- has an address side with enough surface area to fit the delivery address, return address, postage, markings and endorsements, and special service label; and
- is in a box or, if not in a box, is greater than 3/4-inch thick at its thickest point.

This provides mailers many different packaging options for First-Class Mail parcels.

Optional Signature Confirmation Label API Tags

The Signature Confirmation Label API has a set of optional tags that may be sent with your request. Following a brief description of the functions of these tags is a table similar to the *Required Tags* table with values allowed.

You may waive the requirement for the addressee to sign for the item by using the <WaiverOfSignature> tag. If selected, the user agrees to allow delivery to be made without obtaining the signature of the addressee or addressee's agent (if delivery employee judges that article can be left in secure location). In this case, the user authorizes the delivery employee's signature to constitute valid proof of delivery.

The user can post-date the Signature Confirmation label up to four days in advance by using the <LabelDate> tag. The package should not be shipped until the label date requested. On this date the USPS will send a manifest to the PTS, which is the USPS national tracking and Signature Confirmation system.

If you have a need to cross-reference information about a shipment using your own tracking or inventory systems, use the <CustomerRefNo> tag. This string will be included in the daily manifest sent to the PTS system. If you are using the client DUNs number described in the

Appendix, *Signature Confirmation PTS Tracking System Extract File Retrieval*, the extract file you receive will include the information you entered in the <CustomerRefNo> tag. The string you enter appears only in the manifest data, not on the label.

You can now request the USPS to alert you when a customer changes their address. This feature is especially useful to ensure correct billing. The words "Address Service Requested" will appear on the shipping label immediately below the return address. *This service is not available with Label Option 4.* By using the <AddressServiceRequested> tag, you will receive the following service with Priority Mail (for Package Services fees for Address Service Requested, refer to the Domestic Mail Manual, located at the Postal Explorer web site <http://pe.usps.gov>):

- For 12 months after an address change, the mailpiece is forwarded at no charge. However, a separate notice of the new address is returned to you and an address correction fee is charged.
- For months 13-18 after an address change, the mailpiece is returned with the new address attached at no charge.
- After 18 months, or if undeliverable, the mailpiece is returned with the reason for nondelivery attached at no charge.

You can print the Customer Online Record on a separate page from the shipping label by using the <SeparateReceiptPage> tag. This option is further explained in the *Label Options* section.

If you wish to provide your customers the tracking number of your Signature Confirmation package, four optional tags are used to transmit the number via e-mail. Your customers can use this number to track the package via <http://www.usps.com/shipping/trackandconfirm.htm>. The following is a sample e-mail message transmitted with this feature:

```
From: SignatureConfirmationAlert@USPSshippingapis.com
Sent: Tuesday, June 05, 2001 9:37 AM
To: <RecipientEMail>
CC: <SenderEMail>
Subject: Signature Confirmation Number

Dear <RecipientName>:

Below is the Signature Confirmation number you need to determine the delivery
status of your package.

Signature Confirmation Number 21805213907000634941

Your package is scheduled for shipment on 6/5/01.

This email was automatically generated by the US Postal Service
(www.usps.com) at the shipper's request. Any reply to this email will not be
received by the USPS or shipper. The USPS has not collected or retained any
personally identifying information about you or your purchase from this
email.

Thank you,

<SenderName>
mailto:<SenderEmail>
```

The only tag **required** to use the e-mail feature is <RecipientEMail>. The other three are not required to use the e-mail feature.

- The name of the person or company sending the e-mail is entered with the <SenderName> tag. This name will appear in the text of the Signature Confirmation e-mail message.
- The address of the person or company sending the e-mail is entered with the <SenderEMail> tag. This address will appear in the text of the Signature Confirmation e-mail message. This address will be cc:'d when the e-mail is sent.
- The name of the person or company receiving the e-mail is entered with the <RecipientName> tag. This name will appear in the TO: field of the Signature Confirmation e-mail message as well as in the text of the message.
- The address of the person or company receiving the e-mail is entered with the <RecipientEMail> tag. This e-mail address will appear in the TO: field of the Signature Confirmation e-mail message. Although this field is considered optional for the Signature Confirmation API, if e-mail is desired it is the only required field. Without this field, the e-mail will not be sent.

Input	XML Tag	Values Allowed
No Signature Required for Delivery	<WaiverOfSignature>	Enter "True" if you do not want a signature for receipt of the package or "False" if you do. False is assumed if no value is entered.
Label & Customer Online Record Printed on 2 Pages	<SeparateReceiptPage>	Enter "True" if you want the shipping label and online customer record printed on 2 separate pages or "False" if you want them printed on the same single page. False is assumed if no value is entered.
Mailing Location	<POZipCode>	ZIP Code of Post Office or collection box where item is mailed . May be different than From ZIP Code.
Date of the Label	<LabelDate>	Enter the date in either format: dd-mmm-yyyy, such as 10-Jan-2001, or dd/mm/yyyy, such as 10/01/2001.
User-assigned Number for Internal Use	<CustomerRefNo>	Any combination of alpha and numeric characters can be entered, up to a maximum of 30.
Service Provided when Address Changes	<AddressServiceRequested>	Enter "True" or "False." False is assumed if no value is entered.
Name of E-mail Sender	<SenderName>	The name of the person or company sending the e-mail.
E-mail Address of Sender	<SenderEMail>	Valid e-mail address must be used.
Name of E-mail Recipient	<RecipientName>	The name of the person or company receiving the e-mail.
E-mail Address of Recipient	<RecipientEMail>	Valid e-mail address must be used.

"Sample" Request

When you are initially granted access to the production server, the only output you are able to receive is a "Sample" label. This continues until you receive printer approval from the ICCC.

You should start using new valid data for the sample labels. “Sample” requests **must** be used for all testing purposes. Using “Live” requests enters the package in the USPS tracking system.

All of the test script code contained in this document can be cut and pasted for your use in testing the software. To copy the test script code from this PDF file, click on the icon for “Text Selector” and highlight the code. (The icon will look like



or



depending on your version of Adobe Acrobat.) You can then copy the code and paste it into your test document. Remember, however, that all data and attribute values in this document are for illustration purposes and are to be replaced by your actual values. For instance, a line of sample code may be:

```
<FromName>Joe Smith</FromName>
```

In this instance, you will replace “Joe Smith” with the name of the person sending the package when making your request. **Also remember that you are provided with a different server name to send “live” requests.**

Refer to the *XML Tags and Values Allowed* section, above, to build your “Sample” request. When building the XML request, pay particular attention to the **order and case** for tags. An error message will be returned if an incorrect value is entered. Also, be aware of the maximum character amounts allowed for some tags. If the user enters more than those amounts, an error will not be generated. **The API will simply pass in the characters up to the maximum amount allowed and disregard the rest.** This is important since the resulting value could prevent delivery.



Developers: For sample code utilizing Perl and ASP, refer to the Domestic Rates Calculator API and Track/Confirm API user's guides.

The “Sample” XML request should be in the following form and sequence (optional tags are in bold):

```
<SigConfirmCertifyRequest USERID='xxxxxxxx' PASSWORD='xxxxxxxx'>
  <Option></Option>
  <ImageParameters></ImageParameters>
  <FromName></FromName>
  <FromFirm></FromFirm>
  <FromAddress1></FromAddress1>
  <FromAddress2></FromAddress2>
  <FromCity></FromCity>
  <FromState></FromState>
  <FromZip5></FromZip5>
  <FromZip4></FromZip4>
  <ToName></ToName>
  <ToFirm></ToFirm>
  <ToAddress1></ToAddress1>
  <ToAddress2></ToAddress2>
  <ToCity></ToCity>
  <ToState></ToState>
  <ToZip5></ToZip5>
```

```

<ToZip4></ToZip4>
<WeightInOunces></WeightInOunces>
<ServiceType></ServiceType>
<WaiverOfSignature></WaiverOfSignature>
<SeparateReceiptPage></SeparateReceiptPage>
<POZipCode></POZipCode>
<ImageType></ImageType>
<LabelDate></LabelDate>
<CustomerRefNo></CustomerRefNo>
<AddressServiceRequested></AddressServiceRequested>
<SenderName></SenderName>
<SenderEMail></SenderEMail>
<RecipientName></RecipientName>
<RecipientEMail></RecipientEMail>
</SigConfirmCertifyRequest>

```

“Live” Request

When the ICCC has notified you of your printer approval, full production access is immediately granted.

Refer to the *XML Tags and Values Allowed* section, above, to build your “Live” request. When building the XML request, pay particular attention to the *order and case* for tags. An error message will be returned if an incorrect value is entered. Also, be aware of the maximum character amounts allowed for some tags. If the user enters more than those amounts, an error will not be generated. **The API will simply pass in the characters up to the maximum amount allowed and disregard the rest.** This is important since the resulting value could prevent delivery.

Remember that all data and attribute values in this document are for illustration purposes and are to be replaced by your actual values. For instance, a line of sample code may be:

```
<FromName>Joe Smith</FromName>
```

In this instance, you will replace “Joe Smith” with the name of the person sending the package when making your request. **Also remember that you are provided with a different server name to send “live” requests.**

The “Live” XML request should be in the following form and sequence (optional tags are bold):

```

<SignatureConfirmationRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <Option>3</Option>
  <ImageParameters></ImageParameters>
  <FromName>Joe Smith</FromName>
  <FromFirm>ABC Corp.</FromFirm>
  <FromAddress1>Apt. 3C</FromAddress1>
  <FromAddress2>6406 Ivy Lane</FromAddress2>
  <FromCity>Greenbelt</FromCity>
  <FromState>MD</FromState>
  <FromZip5>20770</FromZip5>
  <FromZip4>1234</FromZip4>
  <ToName>Tom Collins</ToName>
  <ToFirm>XYZ Corp.</ToFirm>
  <ToAddress1>Suite 4D</ToAddress1>
  <ToAddress2>8 Wildwood Drive</ToAddress2>
  <ToCity>Old Lyme</ToCity>

```

```

<ToState>CT</ToState>
<ToZip5>06371</ToZip5>
<ToZip4>5678</ToZip4>
<WeightInOunces>32</WeightInOunces>
<ServiceType>Priority</ServiceType>
<WaiverOfSignature></WaiverOfSignature>
<SeparateReceiptPage></SeparateReceiptPage>
<POZipCode></POZipCode>
<ImageType>TIF</ImageType>
<LabelDate></LabelDate>
<CustomerRefNo></CustomerRefNo>
<AddressServiceRequested></AddressServiceRequested>
<SenderName></SenderName>
<SenderEMail></SenderEMail>
<RecipientName></RecipientName>
<RecipientEMail></RecipientEMail>
</SignatureConfirmationRequest>

```

Visual Basic Request

Using the Microsoft XML object model in Visual Basic, such a request can be built as shown below. In this code sample, the data needed to build the XML is obtained from a form. The True/False type elements are obtained from a check box control, the ServiceType element is obtained from an option button control, and the ImageType is from a combo box control. All other fields are obtained from text box controls.

```

Dim RequestLevel As IXMLDOMElement
Dim DCElementLevel As IXMLDOMElement
Dim inetGet As New clsInetGet
Dim response As String
Dim t As Variant

Dim xmlDoc As New DOMDocument
Set RequestLevel = xmlDoc.createElement("SignatureConfirmationRequest")
RequestLevel.setAttribute "USERID", userid.Text
RequestLevel.setAttribute "PASSWORD", password.Text

'Load Signature Confirmation values into request.

Set DCElementLevel = xmlDoc.createElement("Option")
Set t = xmlDoc.createTextNode(Option.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ImageParameters")
Set t = xmlDoc.createTextNode(ImageParam.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromName")
Set t = xmlDoc.createTextNode(FromName.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromFirm")
Set t = xmlDoc.createTextNode(FromFirm.Text)

```

```
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromAddress1")
Set t = xmlDoc.createTextNode(FromAddr1.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromAddress2")
Set t = xmlDoc.createTextNode(FromAddr2.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromCity")
Set t = xmlDoc.createTextNode(FromCity.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromState")
Set t = xmlDoc.createTextNode(FromState.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromZip5")
Set t = xmlDoc.createTextNode(FromZip5.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("FromZip4")
Set t = xmlDoc.createTextNode(FromZip4.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ToName")
Set t = xmlDoc.createTextNode(ToName.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ToFirm")
Set t = xmlDoc.createTextNode(ToFirm.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ToAddress1")
Set t = xmlDoc.createTextNode(ToAddr1.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ToAddress2")
Set t = xmlDoc.createTextNode(ToAddr2.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ToCity")
Set t = xmlDoc.createTextNode(ToCity.Text)
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)
```

```
Set DCElementLevel = xmlDoc.createElement("ToState")
Set t = xmlDoc.createTextNode(ToState.Text)
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ToZip5")
Set t = xmlDoc.createTextNode(ToZip5.Text)
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ToZip4")
Set t = xmlDoc.createTextNode(ToZip4.Text)
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("WeightInOunces")
Set t = xmlDoc.createTextNode(weight.Text)
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("ServiceType")
Set t = xmlDoc.createTextNode(ServiceType.Text)
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("WaiverOfSignature")
If cbWaiveSig.Value = vbChecked Then
    Set t = xmlDoc.createTextNode("TRUE")
Else
    Set t = xmlDoc.createTextNode("FALSE")
End If
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

Set DCElementLevel = xmlDoc.createElement("SeparateReceiptPage")
If cbSeparateReceipt.Value = vbChecked Then
    Set t = xmlDoc.createTextNode("TRUE")
Else
    Set t = xmlDoc.createTextNode("FALSE")
End If
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

If POZipCode.Text <> "" Then
    Set DCElementLevel = xmlDoc.createElement("POZipCode ")
    Set t = xmlDoc.createTextNode(POZipCode.Text)
    DCElementLevel.appendChild(t)
    Call RequestLevel.appendChild(DCElementLevel)
End If

Set DCElementLevel = xmlDoc.createElement("ImageType")
Set t = xmlDoc.createTextNode(Combo1.Text)
DCElementLevel.appendChild(t)
Call RequestLevel.appendChild(DCElementLevel)

If LabelDate.Text <> "" Then
```

```
    Set DCElementLevel = xmlDoc.createElement("LabelDate")
    Set t = xmlDoc.createTextNode(LabelDate.Text)
    DCElementLevel.appendChild (t)
    Call RequestLevel.appendChild(DCElementLevel)
End If

If CustomerRefNo.Text <> "" Then
    Set DCElementLevel = xmlDoc.createElement("CustomerRefNo")
    Set t = xmlDoc.createTextNode(CustomerRefNo.Text)
    DCElementLevel.appendChild (t)
    Call RequestLevel.appendChild(DCElementLevel)
End If

Set DCElementLevel = xmlDoc.createElement("AddressServiceRequested")
If cbAddressService.Value = vbChecked Then
    Set t = xmlDoc.createTextNode("TRUE")
Else
    Set t = xmlDoc.createTextNode("FALSE")
End If
DCElementLevel.appendChild (t)
Call RequestLevel.appendChild(DCElementLevel)

If SenderName.Text <> "" Then
    Set DCElementLevel = xmlDoc.createElement("SenderName")
    Set t = xmlDoc.createTextNode(SenderName.Text)
    DCElementLevel.appendChild (t)
    Call RequestLevel.appendChild(DCElementLevel)
End If

If SenderEMail.Text <> "" Then
    Set DCElementLevel = xmlDoc.createElement("SenderEMail")
    Set t = xmlDoc.createTextNode(SenderEMail.Text)
    DCElementLevel.appendChild (t)
    Call RequestLevel.appendChild(DCElementLevel)
End If

If RecipientName.Text <> "" Then
    Set DCElementLevel = xmlDoc.createElement("RecipientName")
    Set t = xmlDoc.createTextNode(RecipientName.Text)
    DCElementLevel.appendChild (t)
    Call RequestLevel.appendChild(DCElementLevel)
End If

If RecipientEMail.Text <> "" Then
    Set DCElementLevel = xmlDoc.createElement("RecipientEMail")
    Set t = xmlDoc.createTextNode(RecipientEMail.Text)
    DCElementLevel.appendChild (t)
    Call RequestLevel.appendChild(DCElementLevel)
End If

Call xmlDoc.appendChild(RequestLevel)
Set RequestLevel = Nothing
Set DCElementLevel = Nothing
```

Steps 2 & 3: Make the Internet Connection and Send the XML Request

These two steps are presented together to simplify things. The two steps actually involve four separate functions:

1. making the connection to the USPS Shipping API server (test server or production server)
2. sending the request (whether Visual Basic, Perl, ASP, or any other language)
3. receiving the response from the API server
4. closing the Internet connection

These steps are identical for sending “Canned” test requests, “Sample” requests, or “Live” requests. **Remember, however, that you are provided with a different server name to send “live” requests.**

This section describes two ways to make the Internet connection. This is not an all-inclusive list. It simply represents the most common and easiest ways to make the Internet connection.

- Using the USPS-supplied HTTP Connection DLL

The HTTP Connection DLL is recommended for NT systems. This software, created specifically for the USPS API implementation, provides e-tailers with a thread-safe sockets interface to submit XML requests and receive XML responses from the API server.

- Using Microsoft's WinInet

Although you can use the WinInet DLL to make the connection to the API server, it is not recommended for server applications due to limitations in the DLL. It is recommended that you either use the USPS-supplied HTTP Connection DLL or write your own sockets interface that can be used to make multiple connections and will remain thread-safe.

Using HTTP Connection DLL

To obtain this code you must submit a Licensing Agreement. See the *Administrative Guide for APIs* for the agreement.

Using WinInet

This sample code shows how to use Microsoft's WinInet DLL to make the Internet connection, using either the “GET” or “POST” (necessary for requests over 2K in size) methods. XMLSTRING represents the URL-encoded XML request and SERVERNAME indicates the name of the USPS web site to which you are connecting. The two lines of code presented in boldface below must be changed depending on which step you are completing.

For “Canned” test requests the code should read:

```
File = "/ShippingAPItest.dll?"  
xml = "API=SignatureConfirmation&XML=" & XMLSTRING
```

For “Sample” test requests the code should read:

```
File = "/ShippingAPI.dll?"  
xml = "API=SignatureConfirmationCertify&XML=" & XMLSTRING
```

For "Live" requests the code should read:

```
File = "/ShippingAPI.dll?"  
xml = "API=SignatureConfirmation&XML=" & XMLSTRING
```

Input:

```
Dim hOpen As Long, hConnection As Long, hFile As Long, numread As Long  
Dim File As String, xml As String, sHeader As String, htmlFile As String, tmp  
As String * 2048  
Dim bDoLoop As Boolean
```

```
File = "/ShippingAPI.dll?"  
xml = "API=SignatureConfirmation&XML=" & XMLSTRING
```

```
hOpen = InternetOpen("", 1, vbNullString, vbNullString, 0)
```

```
hConnection = InternetConnect(hOpen, SERVERNAME, 0, _  
    "", "", 3, 0, 0)
```

```
.....  
'get  
'File = File & xml  
'hFile = HttpOpenRequest(hConnection, "GET", File, "HTTP/1.0", vbNullString,  
0, 0, 0)  
'OR  
.....
```

```
.....  
' post  
hFile = HttpOpenRequest(hConnection, "POST", File, "HTTP/1.0", vbNullString,  
0, 0, 0)
```

```
sHeader = "Content-Type: application/x-www-form-urlencoded" _  
    & vbCrLf
```

```
Call HttpAddRequestHeaders(hFile, _  
    sHeader, Len(sHeader), 0)  
.....
```

```
bDoLoop = HttpSendRequest(hFile, vbNullString, 0, xml, Len(xml))
```

```
bDoLoop = True  
While bDoLoop  
    tmp = vbNullString  
    bDoLoop = InternetReadFile(hFile, tmp, Len(tmp), numread)  
    If Not bDoLoop Then  
        Exit Sub  
    Else  
        htmlFile = htmlFile & Left$(tmp, numread)  
        If Not CBool(numread) Then bDoLoop = False  
    End If  
Wend
```

```
If hFile <> 0 Then InternetCloseHandle (hFile)
```

```
If hConnection <> 0 Then InternetCloseHandle (hConnection)
If hOpen <> 0 Then InternetCloseHandle (hOpen)
```

Step 4: Unpack the XML Response

This step is identical for unpacking “Canned” test responses, “Sample” responses, or “Live” responses.

Types of Responses

When the USPS Shipping API returns a response, it will either return a successful response document or an error document. Anytime you receive a response, you should check to see if the document is <Error>. Refer to the *Errors* section.

Using Visual Basic

Using the Microsoft XML object model in Visual Basic, such responses can be unpacked as follows:

```
Dim oChild As IXMLDOMNode
Dim nodeList As IXMLDOMNodeList
Dim i As Integer
Dim SignatureConfirmationLabel As String
Dim SignatureConfirmationNumber As String

xmlDoc.validateOnParse = False
xmlDoc.loadXML (xmlStr) 'Response
Set nodeList = xmlDoc.getElementsByTagName("Error")
If nodeList.length > 0 Then 'Top-level Error
    Set n = nodeList.Item(0)
    For i = 0 To n.childNodes.length - 1
        Set e = n.childNodes.Item(i)
        Select Case e.nodeName
            Case "Source"
            Case "Number"
            Case "Description"
                lblRateMessage.Caption = e.firstChild.nodeValue
            Case "HelpFile"
            Case "HelpContext"
        End Select
    Next i
Else 'no Top-level Error
'Get the list of nodes.
    Set nodeList = xmlDoc.getElementsByTagName
    ("SignatureConfirmationResponse")

    For i = 0 To nodeList.length - 1
        Set oChild = nodeList.Item(i)

        Set oChild = oChild.firstChild
        While Not oChild Is Nothing
            Select Case oChild.nodeName
                Case "SignatureConfirmationLabel"
```

```
        If oChild.hasChildNodes Then
            SignatureConfirmationLabel =
                oChild.firstChild.nodeValue
        Else
            'Err.Raise
            msERR_MISSING_SignatureConfirmationLabel, "",
            msERR_MISSING_SignatureConfirmationLabel
        End If
    Case "SignatureConfirmationReceipt"
        If oChild.hasChildNodes Then
            SignatureConfirmationReceipt =
                oChild.firstChild.nodeValue
        Else
            'Err.Raise
            msERR_MISSING_SignatureConfirmationReceipt, "",
            msERR_MISSING_SignatureConfirmationReceipt
        End If
    Case "SignatureConfirmationNumber"
        If oChild.hasChildNodes Then
            SignatureConfirmationNumber =
                oChild.firstChild.nodeValue
        Else
            'Err.Raise
            msERR_MISSING_SignatureConfirmationNumber, "",
            msERR_MISSING_SignatureConfirmationNumber
        End If
    End Select
    Set oChild = oChild.nextSibling
Wend
Next i
End If
Set xmlDoc = Nothing
```

Base64 Decoding in VBScript in an ASP

```
<%@ Language=VBScript %>
<%
Dim dcOutput()
sBase = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
StreamLength = len(SignatureConfirmationImage)

'-----
'Strip cr/lf
dcCopylen = 0
dcCopy = ""
For j = 1 To StreamLength
    If ((Mid(SignatureConfirmationImage, j, 1) <> vbCr) And
Mid(SignatureConfirmationLabel, j, 1) <> vbLf)) Then
        dcCopy = dcCopy & Mid(SignatureConfirmationImage, j, 1)
        dcCopylen = dcCopylen + 1
    End If
Next
'-----

'-----
'Decode bulk of string
```

```
dcOutputlen = 0
ReDim dcOutput((dcCopylen * 3) / 4)
  For j = 1 To dcCopylen - 4 Step 4
    'map "A"-"/" to 0-63
    a = InStr(1, sBase, Mid(dcCopy, j, 1), vbBinaryCompare) - 1
    b = InStr(1, sBase, Mid(dcCopy, j + 1, 1), vbBinaryCompare) - 1
    c = InStr(1, sBase, Mid(dcCopy, j + 2, 1), vbBinaryCompare) - 1
    d = InStr(1, sBase, Mid(dcCopy, j + 3, 1), vbBinaryCompare) - 1
    'decode 0-63 to 0-255
    dcOutput(dcOutputlen) = (a * 4) Or ((b And 48) / 16)
    dcOutput(dcOutputlen + 1) = ((b And 15) * 16) Or ((c And 60) / 4)
    dcOutput(dcOutputlen + 2) = ((c And 3) * 64) Or (d And 63)

    dcOutputlen = dcOutputlen + 3
  Next
'-----
'-----
'Decode last 1-3 characters
a = InStr(1, sBase, Mid(dcCopy, j, 1), vbBinaryCompare) - 1
b = InStr(1, sBase, Mid(dcCopy, j + 1, 1), vbBinaryCompare) - 1
dcOutput(dcOutputlen) = (a * 4) Or ((b And 48) / 16)
If j + 2 <= dcCopylen Then
  c = InStr(1, sBase, Mid(dcCopy, j + 2, 1), vbBinaryCompare) - 1
  dcOutput(dcOutputlen + 1) = ((b And 15) * 16) Or ((c And 60) / 4)
  dcOutputlen = dcOutputlen + 1
End If
If j + 3 <= dcCopylen Then
  d = InStr(1, sBase, Mid(dcCopy, j + 3, 1), vbBinaryCompare) - 1
  dcOutput(dcOutputlen + 2) = ((c And 3) * 64) Or (d And 63)
  dcOutputlen = dcOutputlen + 1
End If

Response.ContentType = "image/tiff"
for j = 0 to dcOutputlen - 1
  response.binarywrite chrB(dcOutput(j))
next
%>
```

Errors

Error conditions are handled at the main XML document level. 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: an API developer sends 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. When parsing, it is best to check for an error document first, before checking for good data.

Error documents follow the Visual Basic error standards and have following format:

```
<Error>
  <Number></Number>
  <Source></Source>
```

```

    <Description></Description>
    <HelpFile></HelpFile>
    <HelpContext></HelpContext>
</Error>

```

where:

- Number = the error number generated by the API server
- Source = the component and interface that generated the error on the API server
- Description = the error description
- HelpFile = [reserved for future use]
- HelpContext = [reserved for future use]

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

Output

After following Technical Step 4 and unpacking the XML response, you will have the output from your request. This section describes the different outputs resulting from “Canned” test requests, “Sample” requests, and “Live” requests. All three types of requests result in an XML response with the following tags:

Output	XML Tag
Type of Response	<SignatureConfirmationResponse>
Signature Confirmation ID Number (PIC #)	<SignatureConfirmationNumber>
Signature Confirmation Label	<SignatureConfirmationLabel>
Signature Confirmation Customer Online Record*	<SignatureConfirmationReceipt>

*The API returns this tag only if the <SeparateReceiptPage> tag was set to TRUE in the request. Otherwise the Customer Online Record is contained within the label image and the <SignatureConfirmationReceipt> tag is NOT present in the response.

“Canned” Test Responses

For your test to be successful, the following responses to Valid Test Requests and Pre-defined Test Requests should be returned *verbatim*. If any values were changes in your request, the following default error will occur:

```

<Error>
  <Number>-2147219040</Number>
  <Source>SOLServerTest;SOLServerTest.DeliveryConfirmationV2_Respond</Source>
  <Description>This Information has not been included in this Test
  Server.</Description>
  <HelpFile></HelpFile>
  <HelpContext></HelpContext>
</Error>

```

Although the input may be valid, the response will still raise this error, because those particular values have not been included in this test server. Refer to the *Errors* section for an explanation of any other returned errors.

Response to Test Request #1

```

<SignatureConfirmationResponse>
  <SignatureConfirmationNumber>22805213907146800684</SignatureConfirmationNumber>

```


. (more data here)

```
.  
MDAwMDAxNzgxOCaWMDAwMCBuDQowMDAwMDAwMDEwIDAwMDAwIG4NCjAwMDAwMTc1  
OTcgMDAwMDAgbg0KMDAwMDAxNzcxMiAwMDAwMCBuDQp0cmFpbGVyDQo8PA0KL1Np  
emUgOA0KL1Jvb3QgMSAwIFINCj4+DQpzdGFydHhyZWYNCjE4MDUzDQolJUVPRg0K  
  </SignatureConfirmationLabel>  
</SignatureConfirmationResponse>
```

Response to Pre-defined Error Request #1

```
<Error>  
  <Number>-2147218903</Number>  
  <Source>SOLServerTest;SOLServerTest.DeliveryConfirmationV2_Respond</Source>  
  <Description>Invalid Service Type.  You sent:  Express</Description>  
  <HelpFile></HelpFile>  
  <HelpContext></HelpContext>  
</Error>
```

Response to Pre-defined Error Request #2

```
<Error>  
  <Number>-2147218901</Number>  
  <Source>SOLServerTest;SOLServerTest.DeliveryConfirmationV2_Respond</Source>  
  <Description>The weight in ounces must be numeric.  </Description>  
  <HelpFile></HelpFile>  
  <HelpContext></HelpContext>  
</Error>
```

Response to Pre-defined Error Request #3

```
<Error>  
  <Number>-2147218900</Number>  
  <Source>SOLServerTest;SOLServerTest.DeliveryConfirmationV2_Respond</Source>  
  <Description>Invalid image type.  </Description>  
  <HelpFile></HelpFile>  
  <HelpContext></HelpContext>  
</Error>
```

Response to Pre-defined Error Request #4

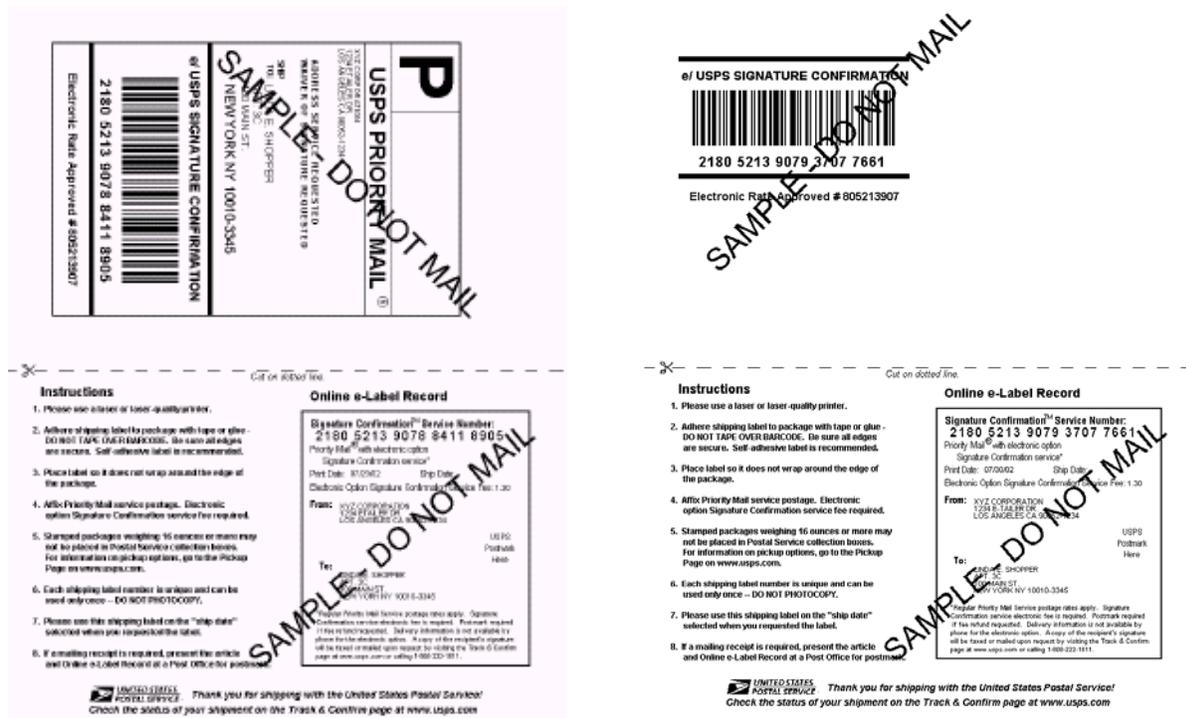
```
<Error>
  <Number>-2147218886</Number>
  <Source>SQLServerTest;SQLServerTest.DeliveryConfirmationV2_Respond</Source>
  <Description>Label date must be within the next seven days.</Description>
  <HelpFile></HelpFile>
  <HelpContext></HelpContext>
</Error>
```

“Sample” Responses

The image returned is Base64-encoded in PDF, JPEG, GIF, or TIF format, according to your request (<ImageType>). It must be decoded before use. For additional information on Base64-encoding and decoding, consult the following URLs:
<http://www.ietf.org/rfc/rfc1421.txt?number=1421> (Section 4.3.2.4) and
<http://www.ietf.org/rfc/rfc2045.txt?number=2045> (Section 6.8).

<p>Important: When printing PDF files with barcodes, be sure that the “Fit to Page” option in the print dialogue box of Adobe Acrobat is unchecked.</p>

Depending on the “Option” included in the request, your output will return one of the following labels. After decoding, the label should look like Label Option 3 (left) or Label Option 4 (right).



(The images above are not to scale.)

“Live” Responses

The image returned is Base64-encoded in PDF, JPEG, GIF, or TIF format, according to your request (<ImageType>). It must be decoded before use. For additional information on Base64-encoding and decoding, consult the following URLs:

<http://www.ietf.org/rfc/rfc1421.txt?number=1421> (Section 4.3.2.4) and

<http://www.ietf.org/rfc/rfc2045.txt?number=2045> (Section 6.8).

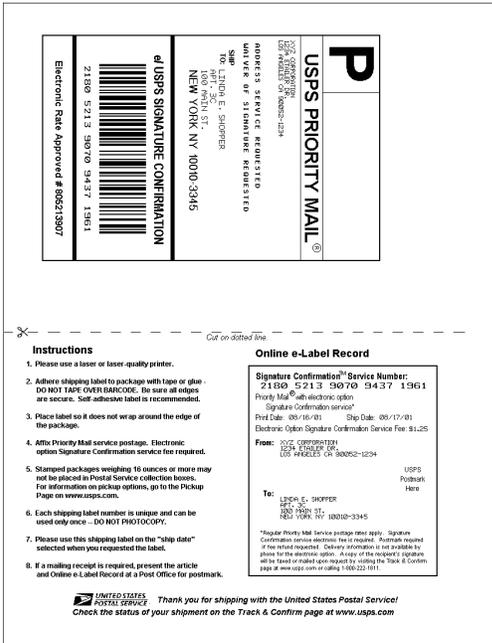
Important: When printing PDF files with barcodes, be sure that the “Fit to Page” option in the print dialogue box of Adobe Acrobat is **unchecked**.

XML Output Example

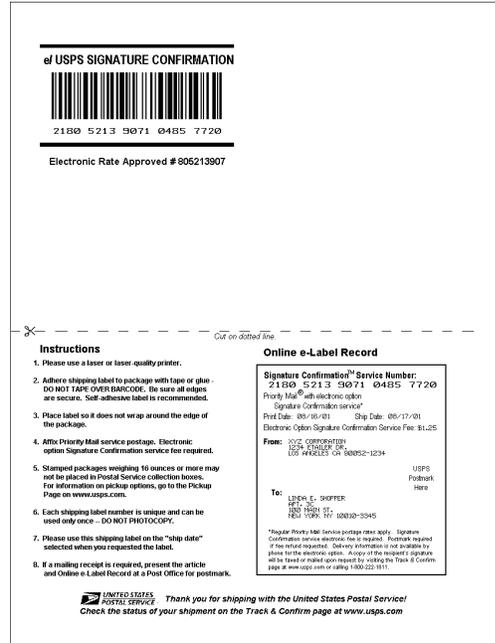
```
<SignatureConfirmationResponse>
  <SignatureConfirmationNumber>02805213907136314906
</SignatureConfirmationNumber>
  <SignatureConfirmationLabel>
SUKqAAgAAAASAP4ABAABAAAAAAAAAAAAAAAAABBAABAAAAVAIAAAEBBAABAAAAkAEAAAIbAwABAAA
AAQAAAAMBawABAAAAABAAAAAYBAwABAAAAAAAAAAAAoBAwABAAAAAGAAABEBBAAAA////////+5D
9gC43ML+v//////// /v/fJev/t9/f//f//+HiCuI+K/jf4j4/2F8/V9DxP///3
. .(more data here) . .
</SignatureConfirmationLabel>
</SignatureConfirmationResponse>
```

Depending on the "Option" included in the request, your output will return the following label:

Label Option 3: The label should be printed on a self-adhesive label at least 5" x 7". After decoding, the label should look like:



Label Option 4: The label should be printed on a self-adhesive label at least 4" x 3". After decoding, the label should look like:



(The images above are not to scale.)

Mailing Instructions and Label Placement on Packages

Third-party providers of labels should be sure to provide label placement information to their end-users. This critical information allows the Postal Service to accurately scan labels. A missing or improperly applied label does not allow the USPS to meet your expectations for providing Signature Confirmation information and will impact your ability to meet the 95% quality threshold. Whenever possible the barcode label should be printed on a self-adhesive label. Provide the following instructions to the end-users of all Signature Confirmation labels:

- Whenever applying a label to a package, you must place it on the address side of the package. USPS preformatted labels must be placed either above the delivery address and to the right of the return address or to the left of the delivery address.
- Do not cover the barcodes with tape or plastic wrap. Barcode scanners cannot read plastic wrap labels that are bent or wrapped around a package.
- If you select Priority Mail service, the parcel weighs 16 ounces or more, and has stamp postage affixed, it must be mailed at a USPS retail window. Stamped packages over 16 ounces may not be placed in USPS collection boxes. For complete information on pickup options, contact your local Postmaster.

Introduction to the Signature Confirmation Barcode Number API

The Signature Confirmation Barcode Number API generates a Signature Confirmation number that can be applied to user-generated labels. This API *only* returns the number, which must be used when exercising the option of generating a custom label.

User ID and Password Restrictions

The user ID and password restrictions for this API are identical to those for the Signature Confirmation Label API. Refer to page 4.

Transaction Procedures

The illustration below shows the transactional flow of information to and from the USPS Signature Confirmation Barcode Number API server.

Signature Confirmation Barcode Number API Server



INPUTS

(via XML Request from Customer to USPS)

From ZIP Code
Recipient ZIP Code
Service Type
Package Weight
Mailing Location (optional)
Waiver of Signature (optional)
Label Date (optional)
Reference Number (optional)
E-mail Information (optional)

SERVER TASKS

Builds XML Response
Tracks Packages
Sends E-mail Notification

OUTPUTS

(via XML Response from USPS to Customer)

Confirmation Number

Technical Steps

Step 1: Build the XML Request

XML Tags and Values Allowed (Required & Optional)

The table below presents the *required* XML input tags for generating both “Sample” and “Live” requests and the restrictions on the values allowed. A second table is also included with *optional* tags for the Signature Confirmation API. An error message will be returned if an incorrect value is entered. Also, be aware of the maximum character amounts allowed for some tags. If the user enters more than those amounts, an error will not be generated. **The API will simply pass in the characters up to the maximum amount allowed and disregard the rest.** This is important since the resulting value could prevent delivery.

Required Signature Confirmation Barcode Number Tags

Input	XML Tag	Values Allowed
"Sample" Request	<SignatureConfirmPICCertifyRequest...	Input tag exactly as presented.
"Live" Request	<SignatureConfirmationPICRequest...	Input tag exactly as presented.
User ID	...USERID="userid"...	Use user ID provided with registration.
Password	...PASSWORD="password">	Use password provided with registration.
From ZIP Code	<FromZip5>	Input tag exactly as presented, not all caps. Maximum characters allowed: 5
To ZIP Code	<ToZip5>	Input tag exactly as presented, not all caps. Maximum characters allowed: 5
Mail Service Type	<ServiceType>	Enter one of the valid entries: "Priority" (for Priority Mail) "First Class" (see below for definition) "Parcel Post" "Bound Printed Matter" "Media Mail" "Library Mail"

First-Class Mail Parcels: For the purposes of adding Signature Confirmation service, a First-Class Mail parcel is defined as any piece that:

- has an address side with enough surface area to fit the delivery address, return address, postage, markings and endorsements, and special service label; and
- is in a box or, if not in a box, is greater than 3/4-inch thick at its thickest point.

This provides mailers many different packaging options for First-Class Mail parcels.

Optional Signature Confirmation Barcode Number Tags

The Signature Confirmation Barcode Number API has a set of optional tags that may be sent with your request. Following a brief description of the functions of these tags is a table similar to the *Required Tags* table with values allowed.

You may waive the requirement for the addressee to sign for the item by using the <WaiverOfSignature> tag. If selected, the user agrees to allow delivery to be made without

obtaining the signature of the addressee or addressee's agent (if delivery employee judges that article can be left in secure location). In this case, the user authorizes the delivery employee's signature to constitute valid proof of delivery. (This tag is included in the Signature Confirmation Barcode Number API because the Waiver of Signature information is included in the manifest data sent to the USPS tracking system.)

The user can post-date the Signature Confirmation label up to four days in advance by using the <LabelDate> tag. The package should not be shipped until the label date requested. On this date the USPS will send a manifest to the PTS, which is the USPS national tracking and Signature Confirmation system.

If you have a need to cross-reference information about a shipment using your own tracking or inventory systems, use the <CustomerRefNo> tag. This string will be included in the daily manifest sent to the PTS system. If you are using the client DUNs number described in the Appendix, *Signature Confirmation PTS Tracking System Extract File Retrieval*, the extract file you receive will include the information you entered in the <CustomerRefNo> tag. The string you enter appears only in the manifest data, not on the label.

If you wish to provide your customers the tracking number of your Signature Confirmation package, four optional tags are used to transmit the number via e-mail. Your customers can use this number to track the package <http://www.usps.com/shipping/trackandconfirm.htm>. The following is a sample e-mail message transmitted with this feature:

```
From: SignatureConfirmationAlert@USPSshippingapis.com
Sent: Tuesday, June 05, 2001 9:37 AM
To: <RecipientEMail>
CC: <SenderEMail>
Subject: Signature Confirmation Number

Dear <RecipientName>:

Below is the Signature Confirmation number you need to determine the delivery
status of your package.

Signature Confirmation Number 21805213907000634941

Your package is scheduled for shipment on 6/5/01.

This email was automatically generated by the US Postal Service
(www.usps.com) at the shipper's request. Any reply to this email will not be
received by the USPS or shipper. The USPS has not collected or retained any
personally identifying information about you or your purchase from this
email.

Thank you,

<SenderName>
mailto:<SenderEMail>
```

The only tag **required** to use the e-mail feature is <RecipientEMail>. The other three are not required to use the e-mail feature.

- The name of the person or company sending the e-mail is entered with the <SenderName> tag. This name will appear in the text of the Signature Confirmation e-mail message.
- The address of the person or company sending the e-mail is entered with the <SenderEMail> tag. This address will appear in the text of the Signature Confirmation e-mail message. This address will be cc:'d when the e-mail is sent.
- The name of the person or company receiving the e-mail is entered with the <RecipientName> tag. This name will appear in the TO: field of the Signature Confirmation e-mail message as well as in the text of the message.
- The address of the person or company receiving the e-mail is entered with the <RecipientEMail> tag. This e-mail address will appear in the TO: field of the Signature Confirmation e-mail message. Although this field is considered optional for the Signature Confirmation API, if e-mail is desired it is the only required field. Without this field, the e-mail will not be sent.

Input	XML Tag	Values Allowed
No Signature Required for Delivery	<WaiverOfSignature>	Enter "True" if you do not want a signature for receipt of the package or "False" if you do. False is assumed if no value is entered.
Mailing Location	<POZipCode>	ZIP Code of Post Office or collection box where item is mailed . May be different than From ZIP Code.
Date of the Label	<LabelDate>	Enter the date in either format: dd-mmm-yyyy, such as 10-Jan-2001, or dd/mm/yy, such as 10/01/01.
User-assigned Number for Internal Use	<CustomerRefNo>	Any combination of alpha and numeric characters can be entered, up to a maximum of 30.
Name of E-mail Sender	<SenderName>	Name of sender.
E-mail Address of Sender	<SenderEMail>	Valid e-mail addresses must be used.
Name of E-mail Recipient	<RecipientName>	Name of recipient.
E-mail Address of Recipient	<RecipientEMail>	Valid e-mail addresses must be used.

“Sample” Request

Unlike the Signature Confirmation Label API, this API does not return a label to you, just a Signature Confirmation number. You must use the number on the label you generate. Until you receive printer approval from the ICCC, “Sample” requests **must** be used for all testing purposes. Even after receiving printer approval, always use the “Sample” requests for testing purposes. Using “Live” requests enters the package in the USPS tracking system.

All of the test script code contained in this document can be cut and pasted for your use in testing the software. To copy the test script code from this PDF file, click on the icon for “Text Selector” and highlight the code. (The icon will look like



or



depending on your version of Adobe Acrobat.) You can then copy the code and paste it into your test document. Remember, however, that all data and attribute values in this document are

for illustration purposes and are to be replaced by your actual values. For instance, a line of sample code may be:

```
<RecipientName>Joe Smith</RecipientName>
```

In this instance, you will replace “Joe Smith” with the name of the person sending the package when making your request. **Also remember that you are provided with a different server name to send “live” requests.**

Refer to the *XML Tags and Values Allowed* section, above, to build your “Sample” request. When building the XML request, pay particular attention to the **order and case** for tags. An error message will be returned if an incorrect value is entered. Also, be aware of the maximum character amounts allowed for some tags. If the user enters more than those amounts, an error will not be generated. **The API will simply pass in the characters up to the maximum amount allowed and disregard the rest.** This is important since the resulting value could prevent delivery.

The “Sample” XML request should be in the following form and sequence (optional tags are in bold):

```
<SignatureConfirmPICCertifyRequest USERID='xxxxxxx' PASSWORD='xxxxxxx'>
  <FromZip5></FromZip5>
  <ToZip5></ToZip5>
  <ServiceType></ServiceType>
  <WaiverOfSignature></WaiverOfSignature>
  <POZipCode></POZipCode>
  <LabelDate></LabelDate>
  <CustomerRefNo></CustomerRefNo>
  <SenderName></SenderName>
  <SenderEMail></SenderEMail>
  <RecipientName></RecipientName>
  <RecipientEMail></RecipientEMail>
</SignatureConfirmPICCertifyRequest>
```

“Live” Request

When the ICCC has notified you of your printer approval, full production access is immediately granted.

Refer to the *XML Tags and Values Allowed* section, above, to build your “Live” request. When building the XML request, pay particular attention to the **order and case** for tags. An error message will be returned if an incorrect value is entered. Also, be aware of the maximum character amounts allowed for some tags. If the user enters more than those amounts, an error will not be generated. **The API will simply pass in the characters up to the maximum amount allowed and disregard the rest.** This is important since the resulting value could prevent delivery.

Remember that all data and attribute values in this document are for illustration purposes and are to be replaced by your actual values. For instance, a line of sample code may be:

```
<RecipientName>Joe Smith</RecipientName>
```

In this instance, you will replace “Joe Smith” with the name of the person sending the package when making your request. *Also remember that you are provided with a different server name to send "live" requests.*

The “Live” XML request should be in the following form and sequence (optional tags are in bold):

```
<SignatureConfirmationPICRequest USERID="xxxxxxx" PASSWORD="xxxxxxx">
  <FromZip5></FromZip5>
  <ToZip5></ToZip5>
  <ServiceType></ServiceType>
  <WaiverOfSignature></WaiverOfSignature>
  <POZipCode></POZipCode>
  <LabelDate></LabelDate>
  <CustomerRefNo></CustomerRefNo>
  <SenderName></SenderName>
  <SenderEMail></SenderEMail>
  <RecipientName></RecipientName>
  <RecipientEMail></RecipientEMail>
</SignatureConfirmationPICRequest>
```

Steps 2 & 3: Make the Internet Connection and Send the XML Request

These two steps are presented together to simplify things. The two steps actually involve four separate functions:

1. making the connection to the USPS Shipping API server (test server or production server)
2. sending the request (whether Visual Basic, Perl, ASP, or any other language)
3. receiving the response from the API server
4. closing the Internet connection

These steps are identical for sending “Sample” requests or “Live” requests. **Remember, however, that you are provided with a different server name to send “live” requests.**

This section describes two ways to make the Internet connection. This is not an all-inclusive list. It simply represents the most common and easiest ways to make the Internet connection.

- Using the USPS-supplied HTTP Connection DLL

The HTTP Connection DLL is recommended for NT systems. This software, created specifically for the USPS API implementation, provides e-tailers with a thread-safe sockets interface to submit XML requests and receive XML responses from the API server.

- Using Microsoft's WinInet

Although you can use the WinInet DLL to make the connection to the API server, it is not recommended for server applications due to limitations in the DLL. It is recommended that you either use the USPS-supplied HTTP Connection DLL or write your own sockets interface that can be used to make multiple connections and will remain thread-safe.

Using HTTP Connection DLL

To obtain this code you must submit a Licensing Agreement. See the *Administrative Guide for APIs* for the agreement.

Using WinInet

This sample code shows how to use Microsoft's WinInet DLL to make the Internet connection, using either the "GET" or "POST" (necessary for requests over 2K in size) methods.

XMLSTRING represents the URL-encoded XML request and SERVERNAME indicates the name of the USPS web site to which you are connecting. The two lines of code presented in boldface below must be changed depending on which step you are completing.

For "Sample" test requests the code should read:

```
File = "/ShippingAPI.dll?"  
xml = "API=SignatureConfirmPICCertify&XML=" & XMLSTRING
```

For "Live" requests the code should read:

```
File = "/ShippingAPI.dll?"  
xml = "API=SignatureConfirmationPIC&XML=" & XMLSTRING
```

Input:

```
Dim hOpen As Long, hConnection As Long, hFile As Long, numread As Long  
Dim File As String, xml As String, sHeader As String, htmlFile As String, tmp  
As String * 2048  
Dim bDoLoop As Boolean
```

```
File = "/ShippingAPI.dll?"  
xml = "API=SignatureConfirmationPIC&XML=" & XMLSTRING  
  
hOpen = InternetOpen("", 1, vbNullString, vbNullString, 0)  
  
hConnection = InternetConnect(hOpen, SERVERNAME, 0, _  
    "", "", 3, 0, 0)  
  
.....  
'get  
'File = File & xml
```

```

'hFile = HttpOpenRequest(hConnection, "GET", File, "HTTP/1.0", vbNullString,
0, 0, 0)
'OR
'.....

'.....
' post
hFile = HttpOpenRequest(hConnection, "POST", File, "HTTP/1.0", vbNullString,
0, 0, 0)

sHeader = "Content-Type: application/x-www-form-urlencoded" _
          & vbCrLf

Call HttpAddRequestHeaders(hFile, _
                          sHeader, Len(sHeader), 0)
'.....

bDoLoop = HttpSendRequest(hFile, vbNullString, 0, xml, Len(xml))

bDoLoop = True
While bDoLoop
    tmp = vbNullString
    bDoLoop = InternetReadFile(hFile, tmp, Len(tmp), numread)
    If Not bDoLoop Then
        Exit Sub
    Else
        htmlFile = htmlFile & Left$(tmp, numread)
        If Not CBool(numread) Then bDoLoop = False
    End If
Wend

If hFile <> 0 Then InternetCloseHandle (hFile)
If hConnection <> 0 Then InternetCloseHandle (hConnection)
If hOpen <> 0 Then InternetCloseHandle (hOpen)

```

Step 4: Unpack the XML Response

This step is identical for unpacking “*Sample*” or “*Live*” responses.

Types of Responses

When the USPS Shipping API returns a response, it will either return a successful response document or an error document. Anytime you receive a response, you should check to see if the document is <Error>. Refer to the *Errors* section.

Using Visual Basic

Using the Microsoft XML object model in Visual Basic, such responses can be unpacked as follows:

```

Dim oChild As IXMLDOMNode
Dim nodeList As IXMLDOMNodeList
Dim i As Integer

```

```
Dim SignatureConfirmationLabel As String
Dim SignatureConfirmationNumber As String

xmlDoc.validateOnParse = False
xmlDoc.loadXML (xmlStr) 'Response
Set nodeList = xmlDoc.getElementsByTagName("Error")
If nodeList.length > 0 Then 'Top-level Error
    Set n = nodeList.Item(0)
    For i = 0 To n.childNodes.length - 1
        Set e = n.childNodes.Item(i)
        Select Case e.nodeName
            Case "Source"
            Case "Number"
            Case "Description"
                lblRateMessage.Caption = e.firstChild.nodeValue
            Case "HelpFile"
            Case "HelpContext"
        End Select
    Next i
Else 'no Top-level Error
'Get the list of nodes.
    Set nodeList = xmlDoc.getElementsByTagName
    ("SignatureConfirmationPICResponse")

    For i = 0 To nodeList.length - 1
        Set oChild = nodeList.Item(i)

        Set oChild = oChild.firstChild
        While Not oChild Is Nothing
            Select Case oChild.nodeName
                Case "SignatureConfirmationNumber"
                    If oChild.hasChildNodes Then
                        SignatureConfirmationNumber =
                        oChild.firstChild.nodeValue
                    Else
                        'Err.Raise
                        msERR_MISSING_SignatureConfirmationNumber, "",
                        msERR_MISSING_SignatureConfirmationNumber
                    End If
                End Select
            Set oChild = oChild.nextSibling
        Wend
    Next i
End If
Set xmlDoc = Nothing
```

Errors

Error conditions are handled at the main XML document level. 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: an API developer sends 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. When parsing, it is best to check for an error document first, before checking for good data.

Error documents follow the Visual Basic error standards and have following format:

```
<Error>
  <Number></Number>
  <Source></Source>
  <Description></Description>
  <HelpFile></HelpFile>
  <HelpContext></HelpContext>
</Error>
```

where:

- Number = the error number generated by the API server
- Source = the component and interface that generated the error on the API server
- Description = the error description
- HelpFile = [reserved for future use]
- HelpContext = [reserved for future use]

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

Output

After following Technical Step 4 and unpacking the XML response, you will have the output from your request. This section describes the different outputs resulting from “*Sample*” requests and “*Live*” requests. Requests result in an XML response with the following tags:

Output	XML Tag
“Sample” Response	<SignatureConfirmPICCertifyResponse>
“Live” Response	<SignatureConfirmationPICResponse>
Signature Confirmation ID Number (PIC #)	<SignatureConfirmationNumber>

Responses

The following XML output example shows the form of both “*Sample*” and “*Live*” responses. The only difference is that the word “Certify” will appear in the first tag of “*Sample*” responses.

“Sample” XML Response

```
<SignatureConfirmPICCertifyResponse>
<SignatureConfirmationNumber>02805213907136314906</SignatureConfirmationNumber>
</SignatureConfirmationPICResponse>
```

“Live” XML Response

```
<SignatureConfirmationPICResponse>
<SignatureConfirmationNumber>02805213907136314906</SignatureConfirmationNumber>
</SignatureConfirmationPICResponse>
```

Generating Your Own Label

Once the API has returned the Signature Confirmation number to you, it can be inserted into your custom label. The label, however, must meet USPS requirements. Refer to USPS Publication 91, Appendix G, at <http://www.usps.com/cpim/ftp/pubs/pub91.pdf> or <http://www.usps.com/cpim/ftp/pubs/pub91/welcome.htm>, which contains the complete specifications for the label that must be met. Also refer to the Domestic Mail Manual (DMM), Sections E120, M012, and S918, located at the Postal Explorer web site <http://pe.usps.gov>.

When generating your own barcode, it is recommended that you use the UCC/EAN Code 128 data format. You may also use the USS Code 128 format, but be aware that the USPS plans to stop accepting this format at some point in the future.

If you request multiple numbers, you must be sure to match up the right number and place it on a label and package with the correct from and to address information used on the input request.

The following samples are presented for use in designing your labels to meet USPS specifications. The label on the left should be printed on a self-adhesive label at least 5" x 7". The label on the right should be printed on a self-adhesive label at least 4" x 3". Self-adhesive labels are recommended, but not mandatory.

P	
USPS PRIORITY MAIL®	
XYZ CORPORATION 1234 ETAILER DR. LOS ANGELES CA 90052-1234	
ADDRESS SERVICE REQUESTED WAIVER OF SIGNATURE REQUESTED	
SHIP TO: LINDA E. SHOPPER APT. 3C 100 MAIN ST. NEW YORK NY 10010-3345	
e/ USPS SIGNATURE CONFIRMATION	
	
2180 5213 9071 1534 3489	
Electronic Rate Approved # 805213907	

e/ USPS SIGNATURE CONFIRMATION

2180 5213 9071 2582 9249
Electronic Rate Approved # 805213907

(The images above are not to scale.)

USPS Barcode Requirements

Text

Bold text, placed no less than 0.125 inch and no more than 0.5 inch above the barcode must read "e/ USPS SIGNATURE CONFIRMATION." The minimum size of this text is 12-point sans-serif type. Larger text is preferred but should not exceed the length of the barcode. The text may be abbreviated to read "e/ USPS SIGNATURE CONFIRM" when full text in larger point type would be longer than the barcode symbol.

Numbers

A human readable numeric representation of the barcode must appear no less than 0.125 inch and no more than 0.5 inch below the barcode. It must be in bold sans-serif type and no less than 10 point. Parsing of the human readable numbers should be in groups of four to facilitate manual entry when required.

Identification Bars

Bold horizontal lines at least 0.062 inch thick must appear between 0.125 inch and 0.5 inch above and below the human readable text and numbers to segregate the Signature Confirmation barcode from other information on the shipping label. The length of the line must be equal to the length of the barcode.

For the Signature Confirmation program, human readable information, including the Package Identification Code (PIC), must meet the dimensional requirements below.



(The image above is not to scale.)

Human-Readable Description of Mailpiece Specifications

- The text above the barcode must read: "e/ USPS SIGNATURE CONFIRMATION".
- The font must be sans serif bold, and a minimum of 12 point (14 preferred).
- When 14-point type is used, you may abbreviate the text to read: "e/ USPS SIGNATURE CONFIRM".

- Do not cover the barcodes with tape or plastic wrap. Barcode scanners cannot read plastic wrap labels that are bent or wrapped around a package.
- If you select Priority Mail service, the parcel weighs 16 ounces or more, and has stamp postage affixed, it must be mailed at a USPS retail window. Stamped packages over 16 ounces may not be placed in USPS collection boxes. For complete information on pickup options, contact your local Postmaster.

Appendix: Signature Confirmation Product Tracking System Extract File Retrieval

Delivery information regarding all barcodes sent to you via the Signature Confirmation APIs can be accumulated into a single comma-delimited file to use in your internal systems. This file is available from the Confirmation Services group within the Expedited/Package Services Department of the USPS. It is referred to as the “extract file” from the USPS Product Tracking System (PTS).

The PTS extract file includes information regarding the 20-character barcode Package ID Codes (PICs), such as destination ZIP Code, event code (delivery, manifest, etc), ZIP Code of scan or event, date and time of scan or event, etc. If your company needs a single report of the status of all packages shipped each day, this is where you can retrieve it.

To gain access to the system that creates and stores the extract file for you, you will need to complete some forms. This Appendix provides the forms and instructions needed for approval to retrieve this file from the USPS's PTS. After you have access to the PTS system for your extract files, you will need to contact the ICCC by phone at (800) 344-7779. The ICCC will need your DUNS number to associate it with all PICs created by your company. When the ICCC makes the association (may need at least one working day to complete), you will begin seeing extract files in your FTP area on the PTS system (files are maintained for 30 days before being deleted).

The following is a checklist of activities necessary for gaining access to the PTS extract file:

1.	Obtain a D-U-N-S [®] Number by contacting Dun & Bradstreet at (800) 333-0505 or by accessing their web site at www.dnb.com . Your company may already have a D-U-N-S [®] number and in this case you will need to locate it before continuing with this process.
2.	Complete the U.S. Postal Service Confirmation Services Form 5051, Confirmation Services, Electronic Option Application and PS Form 1357-S, Request for Computer Access (forms included on following pages, also available at: http://www.usps.com/forms/_pdf/ps5051.pdf and http://www.usps.com/forms/_pdf/ps1357s.pdf).
3.	Fax PS Forms 5051 and PS Form 1357-S to: USPS Expedited/Package Services (202) 268-7596 IMPORTANT: Make sure you have included an 8 (or more) alphanumeric character password in field 3 of PS Form 5051. NOTE: The original PS Form 1357-S must also be mailed to: Expedited/Package Services United States Postal Service 475 L'Enfant Plaza, SW RM 4200NB Washington, DC 20260-4299
4.	Receive logon information from the USPS via Post _e CS (information regarding USPS secure document delivery service (Post _e CS) is at http://www.usps.com/postecs/).
5.	Retrieve documentation regarding the extract file, how to read the file, etc., from USPS Publication 91 in HTML format at: http://www.usps.com/cpim/ftp/pubs/pub91/welcome.htm or in PDF format at: http://www.usps.com/cpim/ftp/pubs/pub91.pdf . The Appendices in this document will provide file layout, event codes, and trouble-shooting guides which will help in processing the extract file.

Problems with this process or with logon information may be discussed at USPS Technical Support (877) 264-9693.

USPS Web Tool Kit User's Guide

(SHADED BOXES FOR U.S. POSTAL SERVICE USE ONLY)

Original PS 1357-S Located at:

U.S. Postal Service

Logon ID Assigned

Request for Computer Access
(Instructions for completing form on reverse side.)

Section A: User Identification

1. Requestor's Name (NOT A COMPANY NAME)	2. Social Security No. (DUNS NUMBER)	3. BA Code N/A	4. Finance No. N/A
5. Requestor's Job Title (Title of above person in company)	6. Employment Status (Check One) <input type="checkbox"/> Contractor <input type="checkbox"/> Temporary <input type="checkbox"/> Career <input type="checkbox"/> Casual <input type="checkbox"/> Foreign <input checked="" type="checkbox"/> Nonpostal		
7. USPS Organization/Department N/A	8. USPS or Company Mailing Address (Include (ZIP + 4) Company Name:		
9. PEN Telephone No. Commercial Telephone No.			
10. MSC Code and Name N/A			

11. User responsibility Agreement Statement
I am responsible for Logon/Logoff, all actions pertaining to the use of my assigned logon ID, and will not provide my logon ID to another person. I agree that access to computer data or files not authorized to me is prohibited. I understand my Logon ID may be suspended indefinitely if I violate security procedures or fail to provide update information for Section A whenever I change job positions. I agree that misuse of a USPS computer system may result in disciplinary action and/or criminal prosecution. I understand that any detected misuse of a computer system will be reported to the Inspection Service.

(Read Privacy Act Statement on reverse side before signing. Note: Privacy Act Statement MUST be on reverse side of this form.)

Signature

Date

12. Manager Responsibility Agreement Statement
I agree that modifications to existing service agreements will require additional Form 1357-S requests. I agree that this logon ID will be used for authorized USPS work within the scope of my organization. I also agree that upon termination or transfer of the user, I will advise the Computer Systems Security Officer in writing as to the disposition of the computer files and/or data and logon ID. I will periodically review the use of the assigned logon ID and computer files and/or data.

Manager's Name (Please Print)	Social Security No.	Signature	Date	PEN/Commercial Phone No.
-------------------------------	---------------------	-----------	------	--------------------------

Section B: Computer Access Requested

13a. Describe Support Required
 Logon ID (Select One): New Change Delete Facility Where Access is Required: San Mateo Data Center
 DDE/DR Access Code: _____ User Type: _____ List Existing Logon IDs & Facilities: _____

Access for Delivery Confirmation System
PPP Dial-up Account

13b. Resource Name (Additional room is available on the reverse side)	13c. Sensitive or Proprietary	13d. Access Level Required (See Instructions)

Section C: Computer Access Approvals

14. Contractor Information (Must be completed by the Contracting Officer's Representative (COR))

a. Does the Contract Contain Provisions for (Select Yes or No):	b. Contract Number: _____
1. Screening? <input type="checkbox"/> Yes <input type="checkbox"/> No	c. Contract Expiration Date: _____
2. Security of Information? <input type="checkbox"/> Yes <input type="checkbox"/> No	d. USPS Organization /Department: _____
3. Privacy of Information? <input type="checkbox"/> Yes <input type="checkbox"/> No	
4. Contractor Screening by the Inspection Service? <input type="checkbox"/> Yes <input type="checkbox"/> No	

15. USPS COR's Name (Please Print)	Signature	Date	PEN/Commercial Phone No.
16. Functional System Coordinator's Name (Please Print)	Signature	Date	PEN/Commercial Phone No.
17. Logon ID Administrator's Name (Please Print)	Signature	Date	PEN/Commercial Phone No.

USPS Web Tool Kit User's Guide

U. S. Postal Service

**Confirmation Services - Electronic Option Application
(Web Tools API User)**

Note that grey fields do not apply to the Web Tools program.

Customer Information		
1. Company Name	2. Dun & Bradstreet Number (DUNS)	3. 8 or More Character Alphanumeric Password for PostCS™
4. Address	5. City, State, ZIP+4*	6. Estimated Start Date

*It is the responsibility of the mailer to transmit electronic files that correspond to the actual mail pieces within 24 hours of physically tendering the mail to the USPS. Publication 91, Confirmation Services Technical Guide applies. These electronic files must be accepted into the USPS database to provide delivery status to the mailer. Publication 91 online: www.usps.gov/cpm/ftp/pubs/pub91.pdf

7. Phone Number and Ext.	8. Fax Number	9. Email Address
10. Approving Official's Name and Title (Please Print) <input type="checkbox"/> Mr. <input type="checkbox"/> Ms.	11. *Signature	12. Date

**(I have read the above and agree to provide electronic files for Delivery Confirmation™ and/or Signature Confirmation™. I will comply with all requirements of Publication 91, Confirmation Services Technical Guide.)*

Technical Information

13. Method customer will use to transmit files: <small>Note: For USPS Web Tools users, the preferred method of file transmission is Internet FTP.</small>		
a. <input type="checkbox"/> Internet FTP	b. <input type="checkbox"/> FTP Dial-up (modem) <input type="checkbox"/> FTP Dial-up files will be sent <input type="checkbox"/> Zipped <input type="checkbox"/> UnZipped	c. <input type="checkbox"/> No Transmission from this site (If checked skip to step #15.)

14. **Note: This is the same person who signed PS Form 1357-S and will receive Logon ID/Password for file transmission.		
**IT Manager's Name (Please Print) <input type="checkbox"/> Mr. <input type="checkbox"/> Ms.	Phone Number/Ext. and Fax Number	Email Address
**Shipping Manager's Name (Please Print) <input type="checkbox"/> Mr. <input type="checkbox"/> Ms.	Phone Number/Ext. and Fax Number	Email Address

15. Will commercial vendor software be used to produce electronic file and/or labels?	
a. <input type="checkbox"/> Yes - If yes, please provide the following information:	b. <input type="checkbox"/> No (If checked skip to step #16.)
Software Company Name	Product Name and Version Number
Contact Name and Title <input type="checkbox"/> Mr. <input type="checkbox"/> Ms.	Phone Number

16. Packaging: (Select all that apply.)	17. Type of Mail: (Select all that apply.)	18. Payment of postage type: (Select all that apply.)	19. Estimated mail quantity per week.
a. <input type="checkbox"/> Customat Provided b. <input type="checkbox"/> USPS Provided c. <input type="checkbox"/> None	a. <input type="checkbox"/> Priority Mail d. <input type="checkbox"/> Standard Mail b. <input type="checkbox"/> Package Services c. <input type="checkbox"/> Other	a. <input type="checkbox"/> Stamps b. <input type="checkbox"/> Meter c. <input type="checkbox"/> Manifest d. <input type="checkbox"/> Other	

20. What barcode labels will customer use?		21. Signature Confirmation Customer Printed		22. Integrated barcode	
a. <input type="checkbox"/> Delivery Confirmation Preprinted**	b. <input type="checkbox"/> Signature Confirmation Preprinted**	c. <input type="checkbox"/> Signature Confirmation Customer Printed	d. <input type="checkbox"/> Delivery Confirmation Customer Printed	e. <input type="checkbox"/> Express Mail Manifesting	f. <input type="checkbox"/> Merchandise Return
**Preprinted labels should be sent to:		Phone Number and Extension			
Address		City, State, ZIP+4			

21a. Are you a consolidator? Yes <input type="checkbox"/> No <input type="checkbox"/>	21b. If yes, are you using vendor software? Yes <input type="checkbox"/> No <input type="checkbox"/>	22. Are you a vendor? Yes <input type="checkbox"/> No <input type="checkbox"/>
---	--	--

Optional Customer Information

23.

To Be Completed By USPS Representative

24. Postal Representative		
Name and Title <input type="checkbox"/> Mr. <input type="checkbox"/> Ms.	Phone Number and Extension	Fax Number
Email Address	Pager Number	Area and District
Address	City, State, ZIP+4	Customer Account is: <input type="checkbox"/> National <input type="checkbox"/> Premier <input type="checkbox"/> Other: