ENDICIA LABEL SERVER™

Production updated July 6, 2010

Version 5.0 (Build 3826)

Copyright © 2004-2010 Endicia, A Newell Rubbermaid Company
All rights reserved.
This document is the property of Endicia, A Newell Rubbermaid Company and may not be reproduced, published, transmitted to other parties, in whole or in part, without written permission.

Endicia, A Newell Rubbermaid Company
247 High Street
Palo Alto CA 94301-1041

Table of Contents

1. Overview **5**
About the Endicia Label Server 5
Endicia Label Server Application Programming Interface 5
Web Service System Requirements 6
A Quick Start with the Endicia Label Server Web Service 6
Endicia Label Server Web Service Concepts 6
Security Model 7
About Printing Labels 7
About Setting Up Accounts 7
About Obtaining Pass Phrases 8
Web Passwords 8
Typical Partner Path to Develop and Deploy the Endicia Label Server 9
Endicia Label Server Production Startup Steps 9
Refund Policy 10
About Signing Up Accounts 11
Contact the Endicia Label Server Team 12
Current Changes 12
2. The Endicia Label Server Web Service **14**
Web Methods 14
Using the Endicia Label Server Web Service for the First Time 15
3. Requesting a Shipping Label API (Required) **16**
GetPostageLabel 16
LabelRequest XML Syntax 16
LabelRequestResponse XML Syntax 39
4. Buying Postage API (Required) **43**
RecreditRequest 43
RecreditRequest XML Syntax 43
RecreditRequestResponse XML Syntax 45
Postage Purchases 46
Endicia Bank-to-Bank Postage Payment Options 46
5. Changing the Pass Phrase API (Required) **48**
ChangePassPhraseRequest 48
ChangePassPhraseRequest XML Syntax 48
ChangePassPhraseRequestResponse XML Syntax 50
6. Calculating Postage and Fees API (Single) **51**
PostageRateRequest 51
PostageRateRequest XML Syntax 51

PostageRateResponse XML Syntax 57
7. Calculate Postage And Fees API (Multiple) 60
PostageRatesRequest 60
PostageRatesRequest XML Syntax 60
PostageRatesResponse XML Syntax 64
8. Getting Account Status API 68
AccountStatusRequest 68
AccountStatusRequest XML Syntax 68
AccountStatusResponse XML Syntax 69
9. Requesting a Refund API 71
RefundRequest 71
RefundRequest XML Syntax 71
RefundResponse XML Syntax 72
10. Signing Up for an Account API 76
UserSignupRequest 76
UserSignupRequest XML Syntax 76
UserSignupResponse XML Syntax 80
11. Requesting Package Status API 84
StatusRequest 84
StatusRequest XML Syntax 84
StatusResponse XML Syntax 86
12. Requesting Transaction Lists API 89
GetTransactionsListing 89
GetTransactionsListing XML Syntax 89
TransactionsListingResponse XML Syntax 92
13. Using the SCAN Form API 96
SCANRequest 96
SCANRequest XML Syntax 96
SCANResponse XML Syntax 98
14. Requesting a Carrier Pickup API's 101
14a. Scheduling a Carrier Pickup API 102
CarrierPickupRequest 102
CarrierPickupResponse XML Syntax 104
14b. Canceling a Carrier Pickup Request API 107
CarrierPickupCancel 107
CarrierPickupCancelResponse XML Syntax 108
14c. Changing a Carrier Pickup Request API 111
CarrierPickupChange 111
CarrierPickupChangeResponse XML Syntax 113

14d. Scheduling a Multiple Location Carrier Pickup Request API 116
MultiLocationCarrierPickup 116
MultiLocationCarrierPickupResponse XML Syntax 116
15. Address Validation–Dial-A-ZIP 120
16. Printing using EPL2 and ZPLII 121
17. Miscellaneous Notes and Features 124
Bulk Acceptance Scans (SCAN) 124
Automated Carrier Pickup 124
Endicia Printable Postage XML 124
APO/FPO Addresses 124
Proxy Server 124
Accessing the Endicia Label Server Web Service through SSL/HTTPS 125
Eligible Matter for Delivery or Signature Confirmation 125
Express Mail Sunday and Holiday Service 125
Rubber Stamps 126
18. Frequently Asked Questions 128
19. Mailpiece Shape Chart 131
Appendix A. Change History 133
Appendix B. Error Codes 151
Appendix C. More Source Examples 157
Caution Statements 157
Coldfusion Examples 158
C# Examples 160
Perl Examples 165
PHP Examples 167
Ruby Examples 192
VisualBasic.NET Examples 194
Appendix D. Country Code Table 196
Appendix E. Country and Locality Listing 202
Glossary 206
Index 207

1. OVERVIEW

This Overview describes:

- Endicia Label Server
- Endicia Label Server Application Programming Interface
- Endicia Label Server Web Service System Requirements
- A Quick Start with the Endicia Label Server Web Service
- Endicia Label Server Web Service Concepts
- Security Model of the Endicia Label Server
- Printing Labels
- Setting Up Accounts
- Obtaining Pass Phrases
- Web Passwords
- Typical Partner Path to Develop and Deploy the Endicia Label Server
- Endicia Label Server Production Startup Steps
- Refund Policy
- Account Sign-ups
- Contacting the Endicia Label Server Team
- Current Changes

About the Endicia Label Server

The Endicia Label Server application integrates into a number of applications, from online shopping carts, Web sites, and fulfillment or distribution centers to multiple location, multiple user, and multiple carrier applications, allowing integrators to provide postage solutions to users. The Endicia Label Server uses APIs that allow integrators to seamlessly add United States Postal Service (USPS) shipping into their applications without changes to existing workflows. The Endicia Label Server is an XML Web Service that supports domestic mail classes, PO Boxes, APO/FPO addresses, discounted domestic and international mail with automatically filled customs forms.

The Endicia Label Server provides users with Dial-A-ZIP, Endicia's patented address verification technology. Endicia's Dial-A-ZIP API validates address errors (single or batch mode) using the USPS address database. When a submitted address does not match, the system provides an error code that indicates common address syntax errors. The same process also indicates whether the address is business, residential, or rural. Given costly residential or rural surcharges, this information is crucial in determining the most cost-effective shipping carrier.

Endicia Label Server Application Programming Interface

The Endicia Label Server Application Programming Interface (API) allows integrators to embed 100% Web-based, on-demand services of the USPS into their applications and Web sites to let their customers print postal shipping labels directly from their applications. The Endicia Label Server produces an integrated label image, complete with Stealth (hidden) postage, return addresses, verified delivery addresses, and service barcodes (Delivery Confirmation, Signature Confirmation, Certified Mail with optional Electronic Return Receipt, Express Mail, Confirm Services, or customs forms).

Web Service System Requirements

The Endicia Label Server Web Service requires that you have:

- A deployment computer with an Internet connection.
- The ability to send requests using HTTP 1.1 or SOAP 1.1 protocols.
- The ability to parse XML.

The Endicia Label Server Web Service is a standard XML-based Web Service exposed through Web Services Description Language (WSDL). It can be programmatically accessed by any development environment that supports calling an XML Web Service using SOAP or HTTP POST. It has been highly tested and documented for environments using Microsoft Visual Studio .NET tools for Web Service development.

A Quick Start with the Endicia Label Server Web Service

The basic principle is that you send a label request to the Endicia Label Server Web Service in either HTTP POST or SOAP format. The Endicia Label Server Web Service then authenticates your Account ID and Pass Phrase and processes the label request. When it has processed the request, it sends the label image it generated, according to the parameters that you specified in your request.

To instantiate the Endicia Label Server Web Service, follow the procedures for invoking or making a reference to an XML Web Service in your preferred development environment. After the Endicia Label Server Web Service has been instantiated, you are ready to call its various methods.

The Endicia Label Server Web Service for development purposes (referred to as the Test Server) is at:

https://www.envmgr.com/LabelService/EwsLabelService.asmx

The WSDL (Web Service Description Language) is at:

https://www.envmgr.com/LabelService/EwsLabelService.asmx?WSDL

Upon approval, you will be sent a "Welcome" e-mail message with information for signing up an account and the URL for the Production Server.

Important! Legacy account numbers (DAZzle—PC or Mac) **will not** work with the Endicia Label Server. As required by the USPS, new account numbers must be created that are specific to the Endicia Label Server.

Endicia Label Server Web Service Concepts

The basic concept around the Endicia Label Server is to provide partners, developers, and integrators with the ability to print US Postal Service pre-paid shipping labels with postage.

The interface is a set of Web Services that are designed to be integrated into other applications or Web sites. The Web methods of the Endicia Label Server Web Service allow applications to use the US Postal Service in a manner similar to the shipping APIs that other carriers provide.

Important! There are several differences between using the US Postal Service and other competing carriers (such as FedEx or UPS). Mainly, the differences pertain to the fact that the US Postal Service Shipping Labels include postage, which is treated like money and thus requires certain procedures to safeguard that transaction and allow customers to recover from any mistakes. This differs from other carriers, where a customer may not be billed until a printed label is scanned.

Security Model

The security model of the Endicia Label Server:

- Safeguards the postage.
- Allows Endicia to provide the Endicia Label Server functionality to partners without having the partners go through a security certification.
- Supports labels that have some form of tracking. This means that only mail classes that the USPS scans at some point are supported. This includes all mail classes that include:
    - Delivery Confirmation or Signature Confirmation: First-Class Parcels, Parcel Post, Library Mail, Media Mail, and Priority Mail.
    - Tracking: Express Mail.
    - Customs Form Numbers: International Priority Mail and International Express Mail.
    - PLANET Code: for First-Class Mail.
      Note: The use of PLANET Code requires approval by both USPS and Endicia. For more information, contact the Endicia Label Server team.

About Printing Labels

To print shipping labels, you must set up an account with the US Postal Service through Endicia. Endicia has a secure link with the US Postal Service, and all account information is transferred to them. Additionally, all postage information is also transferred to them electronically. Any postage that is purchased through the Endicia Label Server is transacted through Endicia, but the money never resides at Endicia. Instead, Endicia issues the transactions to take the money from the user's account to the US Postal Service.

About Setting Up Accounts

One of the first things that you must decide is how to set up accounts. Depending on how you are using the Endicia Label Server, each approach will differ.

One fundamental design constraint that impacts how many accounts that you may set up is how the postage is paid for. Currently, Endicia requires that an account can only have one financial way to fund postage. However, a partner can have several accounts, each funded by a different source. Examples of different sources include, bank account, ACH pull or direct debit, ACH push or credit, check, wire transfer, and credit card). For more information about postage payment methods, go to: http://www.endicia.com/SignUp/Notice/PostagePaymentOptions/

There is no limit to the number of accounts that a partner may open for themselves or on behalf of their customers (in the case of some of the Endicia Enterprise Solution customers).

Several different categories of partners use the Endicia Label Server, and depending on their business models, each partner creates accounts in different ways. A few examples follow:

- A partner could be an individual user or company wanting to implement a full Web Service solution into their application. In this case, the partner can set up one Endicia account and allow all transactions to be processed through their application.
- A partner could be an eCommerce Web site that has buyers and sellers. The partner wants to allow the sellers to print shipping labels with postage. In this case, the partner can set up one Endicia account and allow all transactions to flow through that account.
  Exception: If the site is a very high volume site (more than 5 labels per second), you should consider having more than one account to divide the load.
- A partner might have a software product that is sold to other companies. Suppose that this is a large Web-based CRM Enterprise application. In this case, the partner would not want to have the postage liability and instead would assign at least one Endicia account for each one of their customers.
- A partner could be a CRM Enterprise customer (such as a Fortune 500 company) that has several funding mechanisms, and then they should have at least one Endicia account per funding mechanism.
- A partner could be a third-party software provider for multi-carrier shipping systems seeking to integrate PC Postage into their application. In this case, the partner could use either single or multiple accounts depending on their business model. To explore these various options, contact the Endicia Label Server team.

Endicia has created automated interfaces to allow creation of accounts through the API. Currently, partners are provided with a URL to create accounts. For more information, see Account Sign-ups.

About Obtaining Pass Phrases

You must complete the following steps to correctly obtain a production-ready Pass Phrase:

1. During the account setup and sign-up stages, you must select a temporary Pass Phrase on the Web site.
2. After you receive the e-mail message that contains the assigned account number, you must then use the ChangePassPhrase (or ChangePassPhraseXML) Web method of the Endicia Label Server Web Service and create a new Pass Phrase. *This is one of the important elements of the security model as required by the USPS*. If you are catering to Endicia Enterprise Solution customers who each have their own Endicia Account, you must include this functionality into your application so that your customers can set up accounts and then reset their Pass Phrase. This also allows them to independently reset and change their Pass Phrases.

Web Passwords

Endicia also carries the additional concept of the *Web Password*. You use the Web Password to:

- Gain access to the endicia.com Web site.
- View account transactions.
- Change account information.

The Web Password does *not* enable you to buy and print postage.

Typical Partner Path to Develop and Deploy the Endicia Label Server

1. **In Development Mode:**
   a. Build your Endicia Label Server Web Service interface against the Test Server to perform Pass Phrase resets, buy postage, and request shipping labels. In this mode, you can test with any 6-digit account number. (**Note:** Everything that you need to begin developing and testing is outlined in this document.)
2. **To Qualify for Production:**
   a. Complete an Endicia Label Server Agreement (a Label Server Agreement was part of the original email message that was sent to you with this Endicia Label Server documentation).
   b. Complete the required USPS Partner Profile (Exhibit B—Partner Description in the Label Server Agreement).
   c. Obtain approval for your label by sending an email message to the Endicia Label Server Team (LabelServer@endicia.com) a sample label that the Test Server has produced.
3. **To Move to Production:**
   a. Decide how many accounts you need.
   b. Set up one or more accounts (through either the Account Signup API or Endicia.com using a custom link that the Endicia Label Server team has assigned to you).
      i. You must provide a temporary Pass Phrase.
      ii. You will receive an email message with your account number information.
   c. Perform a Pass Phrase Reset using the ChangePassPhrase (or ChangePassPhraseXML) Web method of the Endicia Label Server Web Service. This results in a production Pass Phrase.
   d. Buy postage using the BuyPostage (or BuyPostageXML) Web method of the Endicia Label Server Web Service.
   e. Request a shipping label using the GetPostageLabel (or GetPostageLabelXML) Web method of the Endicia Label Server Web Service.
   Note: You can also request shipping labels with test postage against the production server.

Endicia Label Server Production Startup Steps

After you have been approved to proceed to production, you can do the following:

1. **Open an account:** First, you must open an Endicia postage account. You must link to the following URL:

   http://www.endicia.com/Labelserver/Login.cfm?partid=xxx

   where xxx is the Partner Code assigned to you after completing the steps outlined above.

   The Endicia Label Server Production URL is:

   https://LabelServer.endicia.com

   The preceding URL replaces the development "test" URL that you had been using.

2. **Initialize your account:** As soon as your account is set up, you will receive an e-mail confirmation with the account number. You must then reset your Pass Phrase using the Endicia

Label Server Web Service and set a new Pass Phrase on the account. Successful completion of that step activates the account and allows you to then start buying postage and issuing print postage transactions.

3. **Buy postage:** Verify that you buy postage.

4. **Print postage:** Use the account number and pass phrase to print postage.

5. **Request refunds:** You can request electronic refunds from your account on the Endicia Web site.

6. **Access your account:** Note that you can use your account number and password to access your account at www.endicia.com, and then view your transactions (purchase and print), and also issue refunds.

Refund Policy

Because you are printing shipping labels that have live postage, you must have a mechanism to recover costs if a misprint occurs.

The URL at http://www.endicia.com/ELS/refundpolicy.cfm describes the Endicia Refund Policy. It is current with the USPS rules and regulations. You can either embed it in a frame and display it to your customers or include it in your offering.

Because the Endicia Label Server returns an image of a label, the partner should build in safeguards to allow the end user to make sure that they have successfully printed the label.

Note that because all the labels have some form of tracking, the refunds must be done electronically. Refunds can be requested through an API call or by logging into the Endicia account page on www.endicia.com. No mailed-in physical refunds will be accepted. All electronic refunds must be requested within 10 days of printing the postage. Our system monitors the USPS package tracking system (PTS) and if the label does not appear within 14 (that number could change) days, then the amount of postage is refunded to the account.

The main methods that are available to request refunds are:

- **Manual:** You may log in to the Endicia account on the Web (www.endicia.com) with your account number and pass word and select Submit Refund Request and follow the instructions.
- **The URL method:** This is appropriate for creating one refund through a Web browser that the partner would either integrate into their site using a frame or opening a new window to our site. It is, however, a manual process.
  https://www.endicia.com/ELS/RefundStart.cfm?accountnumber=xxxxxx&webpassword=xxxxxx(&postageidentifier)=xxxxxxxxxxxx

  where:
  - &accountnumber is the users account number
  - &webpassword is the user's Web password and
  - (&postageidentifier) is *one* of three ways to identify the shipment:
    - &transactionid=91xxxxxxxxxxxxxx (picnumber)
    - &pieceid=xxxxxx (piecenumber)

- &customsid=xxxxxxxxxxx (the customs ID number)
  Note: Use &customsid only for international shipments, where the customs ID number has been stored for tracking.

- **The Requesting a Refund API:** For details on how to use the Refund Web Service API for submitting a refund request, refer to Requesting a Refund API.

About Signing Up Accounts

The following are ways to sign up an account.

- **Manually**: This is appropriate for creating one account that the partner would use but can be used to create many accounts. It is, however, a manual process. For example:

  http://www.endicia.com/Labelserver/Login.cfm?partid=xxx

  where xxx is the Partner Code assigned to you, and then follow the screen prompts.

- **Web Service**: The Account Sign-up Web Service allows the partner to embed the account creation parameters completely within their process and user interface and post the results to the Endicia server. The specifications are in the Account Signup API section.

- **Semi-automating the Web sign-up by passing parameters at the URL line**: This allows the partner to provide default values for some of the information on the Account sign-up pages. For example:

  https://www.endicia.com/labelserver/login.cfm?**PartID=XXXX**

  Required Components:

  PartID—Your Partner ID

  Optional Components:

  Company_Name—Your Company Name
  Contact_FName—Your First Name
  Contact_MName—Your Middle Initial
  Contact_LName—Your Last Name
  Contact_Title—Your Title
  Customer_Email—Your Email Address
  Confirm_Email—Your Email Address Again for Verification
  Physical_Address—Your Physical Address
  Physical_City—Your Physical City
  Physical_State—Your Physical State
  Physical_Zip_Code—Your Physical ZIP Code
  Mailing_address—Your Mailing Address
  Mailing_City—Your Mailing Address City
  Mailing_State—Your Mailing Address State
  Mailing_Zip_Code—Your Mailing Address ZIP Code

Putting It All Together:

URL + ? + Component + = + ComponentValue + & + Component2 + = + Component2Value….etc.

The URL with All Information:

https://www.endicia.com/labelserver/login.cfm?PartID=lngh&Company_Name=LengthCo.&Contact_FName=Bob&Contact_MName=D&Contact_LName=Peterson&Contact_Title=CEO&Customer_Email=bob@Lengthco.com&Confirm_Email=bob@Lengthco.com&Physical_Address=123 Frank St.&Physical_City=New York&Physical_Zip_Code=12345

Contact the Endicia Label Server Team

To contact the Endicia Label Server team, send your inquiries through e-mail message to LabelServer@endicia.com. Prior to sending your e-mail messages, review the following:

- Typical Partner Path to Develop and Deploy the Endicia Label Server
- Frequently Asked Questions

Current Changes

Endicia Label Server is backward compatible with previous versions when that is technically feasible. All methods in this document are supported in the production version 5.0 that will be available as of June 29, 2010.

Note: Refer to the Change History appendix for a complete list of changes. Forward this document to your developers, so that they are aware of the new features.

Version 5.0 (Build 3826) contains the following changes, enhancements, and updates:

- Added a Continuation Page to international customs forms to allow up to 30 customs items.
- Added support for integrated labels with customs forms for international, APO/FPO, and other domestic destinations. These labels also allow up to 30 customs items.

Version 4.9 (Build 3741) contains the following changes, enhancements, and updates:

- Added new Web Methods to calculate "all" domestic or international postage rates: CalculatePostageRates and CalculatePostageRatesXML

Version 4.8 (Build 3691) contains the following changes, enhancements, and updates:

- Added support for the updated USPS international and customs forms 2976-A.
- Added element in the LabelRequest XML to specify the EEL/PFC value for the customs forms 2976-A. The Exemption or Exclusion Legend (EEL) or a Proof of Filing Citation (PFC) is required for shipments to an international destination or to an overseas U.S. Territory.

Version 4.7 (Build 3650) contains the following changes, enhancements, and updates:

- Added support for USPS rates effective January 4, 2010 which also includes:
  - New Flat Rate Padded Envelope for Priority Mail Commercial Plus customers.
  - Cubic pricing for qualified Priority Mail Commercial Plus customer. The Label Server automatically selects the lowest rate based on the weight and volume of the mailpiece!

    - Added support for Parcel Select Barcoded Nonpresorted rates.
    - International shipments to Kosovo.

Version 4.5 (Build 3603) contains the following changes, enhancements, and updates:

- Added support for Standard Mail.
- The SundayHolidayDelivery value for the ServiceLevel element in the LabeRequest and PostageRateRequest XMLs is obsolete. It has been replaced with the SundayHolidayDelivery element.

Version 4.4 (Build 3544) contains the following changes, enhancements, and updates:

- Endicia will be generating all label numbers for international mail labels.
- The SCAN form now accepts international label numbers.

Version 4.3 (Build 3485) contains the following changes, enhancements, and updates:

- Added support for USPS NSA rates for qualified accounts.
- As of May 2009, USPS has renamed the regular Priority Mail Flat Rate box to Medium Flat Rate box.

Version 4.2 (Build 3404) contains the following changes, enhancements, and updates:

- Added support for USPS domestic and international rates effective May 11, 2009.
- To comply with USPS requirements, added a FIM to the indicia in EnvelopeSize10 and Postcard size labels.
- The EmailOptions element in the LabelRequest XML is obsolete.
- Added details about purchasing postage and bank-to-bank postage payment options (Buying Postage API).

E-refunds are not currently supported for FCMI through Label Server. We plan to provide support for physical refunds in the next update to Endicia Label Server.

A complete change history for Endicia Label Server can be found in the Change History section.

2. THE ENDICIA LABEL SERVER WEB SERVICE

Web Methods

The following table lists all the Web methods of the Endicia Label Server Web Service:

Web Method Name	Description
GetPostageLabel or GetPostageLabelXML	Request a shipping label.
BuyPostage or BuyPostageXML	Add postage to an account.
ChangePassPhrase or ChangePassPhraseXML	Change Pass Phrase for an account.
CalculatePostageRate or CalculatePostageRateXML	Calculate the postage and fees for a package.
CalculatePostageRates or CalculatePostageRatesXML	Calculate "all" domestic or international postage rates.
GetAccountStatus or GetAccountStatusXML	Request the status of an account.
RefundRequest	Request a refund.
UserSignup	Sign up for an account.
StatusRequest	Request status of a package.
GetTransactionsListing	Request a list of transactions
SCANRequest	Using the SCAN Form
CarrierPickupRequest	Schedule a Carrier Pickup
CarrierPickupCancel	Canceling a Carrier Pickup Request
CarrierPickupChange	Changing a Carrier Pickup Request
MultiLocationCarrierPickup	Scheduling a Multiple Location Carrier Pickup Request

The Endicia Label Server Web Service Web methods can be invoked using one of two protocols, HTTP POST or SOAP (over HTTP).

HTTP POST

The following is a general format of an HTTP POST request and response. The **placeholders** shown in bold font need to be replaced with actual values.

```
POST /LabelService/EwsLabelService.asmx/Web method Name HTTP/1.1
Host: www.envmgr.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

Request content
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

Response content
```

SOAP (Simple Object Access Protocol)

You must install a code library that can call a SOAP-based Web Service. Development tools, such as Visual Studio .Net, are set up to support Web Services out of the box.

Using the Endicia Label Server Web Service for the First Time

Before requesting your first shipping label using the Production Server, you must perform the following steps:

- Sign up for a production account with the URL that will be supplied to you when you show us a test label and have a signed contract.
- Use the ChangePassPhrase (or ChangePassPhraseXML) Web method to do a Pass Phrase "reset" by changing the temporary Pass Phrase assigned to your account during the sign-up process.
- Use the BuyPostage (or BuyPostageXML) Web method to add postage to your account.

Now you are ready to use the GetPostageLabel (or GetPostageLabelXML) Web method to create shipping labels!

The following sections describe the input to, and response from, the Endicia Label Server Web Service Web methods in greater detail.

3. REQUESTING A SHIPPING LABEL API (REQUIRED)

GetPostageLabel

To request a shipping label, use the GetPostageLabel or the GetPostageLabelXML Web method of the Endicia Label Server Web Service.

The following is a sample HTTP POST request for GetPostageLabelXML, which requires a LabelRequest XML string as input. The **placeholders** shown need to be replaced with actual values:

```
POST /LabelService/EwsLabelService.asmx/GetPostageLabelXML HTTP/1.1
Host: www.envmgr.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

labelRequestXML=<LabelRequest> ... </LabelRequest>
```

The response contains a LabelRequestResponse XML string:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<LabelRequestResponse> ... </LabelRequestResponse>
```

LabelRequest XML Syntax

The following table describes the XML tags for the LabelRequest.

Important notes about all elements:

- Omitting any value causes the Endicia Label Server Web Service to use the default value for that element.
- *Italic* items are optional.
- The Endicia Label Server Web Service requires a valid destination address to process a shipping label. For details on address validation, see Miscellaneous Notes and Features.

Table 3-1: LabelRequest XML Elements

Element	Type	Size or Values	Description
LabelRequest	**Root**	**-**	**The root node for the XML request.**
Test	*Attribute*	*YES* *NO*	*Use sample postage for testing.* *Use real postage. (Default)*
LabelType	*Attribute*	*Default* *CertifiedMail* *DestinationConfirm* *Domestic* *International*	*Create label based on mail class. (Default)* *Create Certified Mail label.* *Create PLANET Code label using Destination Confirm service.* ***Note: Use of the DestinationConfirm label type requires special approval from Endicia and the USPS. For more details, contact the Endicia Label Server Team.*** *Create domestic label. Requires use of the LabelSubtype element.* *Create international label.* *When the value of this element is set to Domestic or International, the label will be returned as separate images within the Label node of the LabelRequestResponse XML.*
LabelSubtype	*Attribute*	*Integrated* *None*	*Create an integrated label. The integrated form type must be specified in the IntegratedFormType element.* *No label subtype (Default)* *Required when label type is Domestic.* *If a value for this element is supplied, it must be set to Integrated when label type is Domestic or International,*

Element	Type	Size or Values	Description
LabelSize	*Attribute*		*For Default label type:*
		4×6	*4" W × 6" H (Default)*
		4×5	*4" W × 5" H*
		4×4.5	*4" W × 4.5" H*
		DocTab	*4" W × 6.75" H, Eltron Doc-Tab label*
		6×4	*6" W × 4" H (not available for Express Mail, EPL2 and ZPLII labels)*
			For DestinationConfirm label type:
		7×3	*7" W × 3" H (Default)*
		6×4	*6" W × 4" H*
		Dymo30384	*DYMO #30384 2-part internet label*
		EnvelopeSize10	*#10 Envelope*
		Mailer7x5	*7" W × 5" H*
			For CertifiedMail label type:
		4×6	*4" W × 6" H (Default)*
		7×4	*7" W × 4" H*
		8×3	*8" W × 3" H*
		Booklet	*9" W × 6 "H envelope*
		EnvelopeSize10	*#10 Envelope*

Element	Type	Size or Values	Description
ImageFormat	*Attribute*	*EPL2* *GIF* *JPEG* *PDF* *PNG* *ZPLII*	*EPL2 programming language* *GIF* *JPEG* *PDF* *PNG (Default)* *ZPL II programming language* *Only GIF is available for international labels created using LabelType of Default.* *EPL2 and ZPLII are supported for:* *- Default label type for domestic mail classes.* *- Domestic or International label type when used with integrated form type 2976.* *- International label type when used with Priority Mail International Flat Rate Envelope and Small Flat Rate Box and First Class Mail International.* *The label image returned in the response for EPL2 and ZPLII labels contain binary data, which require special handling. For details, refer to Printing Using EPL2 and ZPLI.*
ImageResolution	*Attribute*	*203* *300*	*203 dpi* *300 dpi* *Image resolution defaults to 300 dpi for domestic labels as well as labels created using LabelType of International.* *Image resolution defaults to 203 for labels with ImageFormat of EPL2 or ZPLII.* *If this element value is supplied when ImageFormat is set to EPL2 or ZPLII, it must be set to 203.*

Element	Type	Size or Values	Description
ImageRotation	*Attribute*	*None* *Rotate90* *Rotate180* *Rotate270*	*No rotation (Default).* *Rotate label image 90 degrees.* *Rotate label image 180 degrees.* *Rotate label image 270 degrees.* *For labels in EPL2 and ZPLII format, selecting Rotate180 prints from top of image buffer; selecting None prints from bottom of image buffer.* *Rotate90 and Rotate270 are not available for labels in EPL2 and ZPLII format.*
RequesterID	Text	50	Requester ID (also called Partner ID) uniquely identifies the system making the request. Endicia assigns this ID. The Test Server does not authenticate the RequesterID. Any text value of 1 to 50 characters is valid.
AccountID	Numeric	6	Account ID for the Endicia postage account. The Test Server does not authenticate the AccountID. Any 6-digit value is valid.
PassPhrase	Text	64	Pass Phrase for the Endicia postage account. The Test Server does not authenticate the PassPhrase. Any text value of 1 to 64 characters is valid.
MailClass	Text	 Express First LibraryMail MediaMail ParcelPost ParcelSelect Priority StandardMail ExpressMailInternational FirstClassMailInternational PriorityMailInternational	<u>Domestic:</u> Express Mail First-Class Mail Library Mail Media Mail Parcel Post Parcel Select Priority Mail Standard Mail <u>International:</u> Express Mail International First-Class Mail International Priority Mail International

Element	Type	Size or Values	Description
DateAdvance	*Numeric*	*0-7*	*The number of days to advance date on the indicium.* *Maximum value: 7 days.*
WeightOz	Numeric	4.1	Weight of the package, in ounces.
MailpieceShape	*Text*	*Card* *Letter* *Flat* *Parcel* *LargeParcel* *IrregularParcel* *OversizedParcel* *FlatRateEnvelope* *FlatRatePaddedEnvelope* *SmallFlatRateBox* *MediumFlatRateBox* *LargeFlatRateBox*	*Shape of the mailpiece.* *If this element is set to either LargeParcel or OversizedParcel, it overrides the value supplied in MailpieceDimensions.* *See the Mailpiece Shape Chart for more information.* *The Flat Rate Padded Envelope is only available to qualifying Priority Mail Commercial Plus customers.The Label Server will return an error if this shape is used with any other pricing.*
MailpieceDimensions	*Node*		*Dimensions of the mailpiece.* *Required for calculating Commercial Plus Cubic pricing and for Priority Mail mailpieces going to Zones 5-8 which are over 1 cubic foot.* *All values must be in inches.*
Length	Numeric	3.3	Length of the mailpiece.
Width	Numeric	3.3	Width of the mailpiece.
Height	Numeric	3.3	Height (or thickness) of the mailpiece.
AutomationRate	*Text*	*TRUE* *FALSE*	*Use applicable automation rate for selected mail class.* *Use retail price. (Default)* *Available only for letter shape mailpiece using First-Class.*

Element	Type	Size or Values	Description
Machinable	*Text*	*TRUE* *FALSE*	*Mailpiece is machinable. (Default)* *Mailpiece is non-machinable.* *If a Parcel Select mailpiece marked as machinable is over 35 lbs. in weight or its MailpieceShape is set to OversizedParcel, it will automatically be charged the non-machinable price.*
ServiceLevel	*Text*	*NextDay2ndDayPOToAddressee*	*Next Day/Second Day Post Office to Addressee Service* *Applies only to Express Mail. See Miscellaneous Notes & Features for more information.*
SundayHolidayDelivery	*Text*	 *TRUE* *FALSE*	*For Express Mail only:* *Request Sunday/Holiday Delivery Service.* *Do not deliver on Sunday or holiday. (Default)*
SortType	*Text*	*BMC* *FiveDigit* *MixedBMC* *Nonpresorted* *Presorted* *SCF* *SinglePiece* *ThreeDigit*	*Sort level for applicable mail classes.* *Required for Parcel Select and Standard Mail.* *Defaults to SinglePiece for mailpieces which do not require a sort type.*
IncludePostage	*Text*	*TRUE* *FALSE*	*Print postage on the label. (Default)* *Do not print postage on the label.*
ReplyPostage	*Text*	*TRUE* *FALSE*	*Print reply postage on the label which means the Sender's and Recipient's address will be swapped when printing.* *Do not print reply postage. (Default)* *Can only be with label type of Default or DestinationConfirm.* *Not available for Express Mail, international mail, integrated labels or when Return Receipt is requested.*

Element	Type	Size or Values	Description
ShowReturnAddress	*Text*	*TRUE* *FALSE*	*Print sender's address on the label. (Default)* *Do not print sender's address on the label.* *Defaults to TRUE for integrated labels.* *Even though this element can be used to hide the return address in the label, the USPS rules requires that a return address must appear on the mailpiece in specific circumstances. For more information, see the USPS Domestic Mail Manual.*
Stealth	*Text*	*TRUE* *FALSE*	*Turn on the use of "stealth" or hidden postage. (Default)* *Turn off stealth.* *Stealth cannot be used with COD, USPS Insurance, Registered Mail, Automation rate, LabelSize of EnvelopeSize10 and Card shape mailpieces.* *For Standard Mail, Stealth is turned on.*
ValidateAddress	*Text*	*TRUE* *FALSE*	*Validate all addresses. (Default)* *Bypass address validation (requires partner to do Address Validation).*
SignatureWaiver	*Text*	 *TRUE* *FALSE*	*For Express Mail only:* *Request waiver of signature for receipt of mailpiece.* *Request signature for receipt of mailpiece. (Default)* *Endicia recommends that the value of this element be set to TRUE.*

Element	Type	Size or Values	Description
NoWeekendDelivery	*Text*	*TRUE* *FALSE*	*For Express Mail only:* *Request that mailpiece should NOT be delivered on a Saturday.* *Request that mailpiece can be delivered on a Saturday (Default)*
Services	*Node*		*Special Services requested for the package.*
CertifiedMail	*Attribute*	*OFF* *ON*	*Certified Mail not requested.* *Certified Mail requested.* *The default value is ON for CertifiedMail label type; otherwise, it is OFF.* *Available for First-Class and Priority Mail.*
COD	*Attribute*	*OFF* *ON*	*COD not requested. (Default)* *COD requested.* *(must affix a completed COD Form 3816 to the mailpiece and take it to the retail USPS counter).*
DeliveryConfirmation	*Attribute*	*OFF* *ON*	*Delivery Confirmation not requested.* *Delivery Confirmation requested.* *The value of this element is automatically set by the Endicia Label Server based on other elements in the LabelRequest XML.* *Do not supply a value for this element. It will be ignored.*

Element	Type	Size or Values	Description
ElectronicReturnReceipt	*Attribute*	*OFF* *ON*	*Electronic Return Receipt not requested. (Default)* *Electronic Return Receipt requested.* *To receive Electronic Return Receipt delivery information, the partner must be registered as a Bulk Proof of Delivery Client with the USPS and develop the software to download the extract file from USPS. The Mailer ID assigned by the USPS for this purpose must be specified in the BpodClientDUNSNumber element.*
InsuredMail	*Attribute*	*OFF* *ON* *UspsOnline* *Endicia*	*No insurance requested. (Default)* *USPS Insurance requested (must affix a completed Form 3813 or 3813-P to the mailpiece and take it to the retail USPS counter).* *USPS Online Insurance requested.* *Endicia Insurance requested (Maximum insurable value: $10,000)* *USPS Online Insurance is available only for mailpieces with Delivery or Signature Confirmation.* *USPS insurance is not allowed for International Mail or when Stealth or ReplyPostage is set to TRUE.* *Endicia insurance fee is not included in the postage price. It is billed to your account.*
RestrictedDelivery	*Attribute*	*OFF* *ON*	*Restricted Delivery not requested. (Default)* *Restricted Delivery requested.*

Element	Type	Size or Values	Description
ReturnReceipt	*Attribute*	*OFF* *ON*	*Return Receipt not requested. (Default)* *Return Receipt requested (must affix a completed Return Receipt Form 3811 to the mailpiece and take it to the retail USPS counter).* *Can only be used with label type of Default or DestinationConfirm.* *Not supported for International Mail.*
SignatureConfirmation	*Attribute*	*OFF* *ON*	*Signature Confirmation not requested. (Default)* *Signature Confirmation requested.* *Can only be used with label type of Default.*
TrackingNumber	*Text*	*22* *14* *12*	*Tracking number to use:* *PIC* *PLANET Code, 14-digit format* *PLANET Code, 12-digit format* *This element must not be supplied when label type is Domestic.* *Not used for international mail.* ***Note: Use of the DestinationConfirm label type requires special approval from Endicia and the USPS. For further details, contact the Endicia Label Server Team.***
CostCenter	*Numeric*	*8*	*Cost-center code for accounting purposes.*
Value	*Currency*	*5.2*	*Value of the mailpiece.* *When a customs form is requested and the individual customs item elements are not supplied, then this value should be set to the total customs value..* *Ignored when CustomsInfo is supplied.*

Element	Type	Size or Values	Description
InsuredValue	*Currency*	*5.2*	*Insured value of the mailpiece.* *Required if insurance or COD is requested for the mailpiece.*
CODAmount	*Currency*	*5.2*	*COD amount to collect.* *Required if COD is requested for the mailpiece.*
Description	*Text*	*50*	*Description of the item shipped. Used for authentication by recipient.* *When requesting an International Mail label or a customs form, a value is required if the LabelRequest XML does not contain any customs declaration elements.*
IntegratedFormType	*Text*	*Form2976* *Form2976A*	*Form 2976* *Form 2976-A* *Required when label subtype is Integrated.*
CustomsFormType	*Text*	*None* *Form2976* *Form2976A*	*No Customs Form (Default)* *Same as CN22* *Same as CP72* *Used for APO/FPO and other destinations as required by USPS regulations.* *Do not use with integrated labels or international mail.*
CustomsFormImageFormat	*Text*	*GIF* *JPEG* *PDF* *PNG*	*GIF* *JPEG* *PDF* *PNG (Default)* *Do not use with integrated labels or international mail.*
CustomsFormImageResolution	*Text*	*300*	*300 dpi (default)* *Do not use with integrated labels or international mail.*

Element	Type	Size or Values	Description
OriginCountry	*Text*	*50*	*Country of Origin of the item. This must be a valid, USPS recognized country.* *Required for International Mail items or when requesting a customs form.* *Ignored when CustomsInfo is supplied.*
ContentsType	*Text*	*Documents* *Gift* *Merchandise* *Other* *ReturnedGoods* *Sample*	*Category of the customs items. Used in customs forms and international labels.* *Default value is Other.* Value of ReturnedGoods is *not available for Priority Mail International Flat Rate Envelope and Customs Form 2976* *Ignored when CustomsInfo is supplied.*
ContentsExplanation	*Text*	*25*	*Explanation of the customs items. Used in customs forms and international labels.* *Required if ContentsType is Other.* *Ignored when CustomsInfo is supplied.*
NonDeliveryOption	*Text*	*Abandon* *Return*	*Non-delivery instructions.Used in customs forms and international labels.* *Treat as abandoned.* *Return to sender. (Default)* *Ignored when CustomsInfo is supplied.*
ReferenceID	*Text*	*50*	*A reference value for logs.*
PartnerCustomerID	Text	25	A unique identifier for the partner's end-user printing the label
PartnerTransactionID	Text	25	A unique identifier for the partner's end-user's transaction such as invoice, transaction number, etc.

Element	Type	Size or Values	Description
BpodClientDunsNumber	*Numeric*	*9*	*Mailer ID of the partner assigned by USPS.* *Allows the partner to receive Electronic Return Receipt information from USPS in an extract file uniquely generated for them, provided they have established themselves with USPS as a Bulk Proof of Delivery Client.*
RubberStamp1	*Text*	*50*	*User-supplied text to print on the label.* *See Miscellaneous Notes & Features, for supported label types and sizes.*
RubberStamp2	*Text*	*50*	*User-supplied text to print on the label.* *See Miscellaneous Notes & Features, for supported label types and sizes.*
RubberStamp3	*Text*	*50*	*User-supplied text to print on the label.* *See Miscellaneous Notes & Features, for supported label types and sizes.*
EntryFacility	*Text*	*DBMC* *DDU* *DSCF* *OBMC* *Other*	*Postal facility where the mail is entered. If this element is not set to Other, the ZIP Code of this facility must be specified in POZipCode.* *Required for Parcel Select and Standard Mail.* *Destination BMC* *Destination Delivery Unit* *Destination Sectional Center Facility* *Origin BMC* *Other postal facility (Default)*

Element	Type	Size or Values	Description
POZipCode	*Text*	*5*	*ZIP Code of Post Office or collection box where item is mailed. May be different than FromPostalCode.* *Used for determining the zone and calculating the postage price.* *Required when EntryFacility is not set to Other. The value of this element must contain the zip code of the postal facility specified in EntryFacility.*
ShipDate	*Date*	*MM/DD/YYYY*	*Date mailpiece is shipped.* *Required for Express Mail Sunday/Holiday Delivery Service. Ignored for other mail classes.*
ShipTime	*Time*	*HH:MM AM or HH:MM PM*	*Time mailpiece is shipped.* *Applies only to Express Mail Sunday/Holiday Delivery Service. Ignored for other mail classes.* *If this element is not supplied, it defaults to 12:01 AM.*
EelPfc	*Text*	*35*	*Exemption or Exclusion Legend (EEL) or a Proof of Filing Citation (PFC).Required for shipments to an international destination or to an overseas U.S. Territory.* *It is recommended to supply a value for this element as the USPS will likely require it in the near future.*
CustomsCertify	*Text*	*TRUE* *FALSE*	*TRUE means the customs information is certified to be correct and the CustomsSigner name should be printed.*

Element	Type	Size or Values	Description
CustomsSigner	*Text*	*47*	*Name of person certifying that the customs information is correct. This name prints on the customs form in place of a signature if CustomsCertify is TRUE.* *Required if CustomsCertify is TRUE.*
ResponseOptions	*Node*		*Optional XML elements to include in the LabelRequestResponse.*
PostagePrice	*Attribute*	*TRUE* *FALSE*	*TRUE means the response contains the PostagePrice node.*
FromName	*Text*	*47*	*Sender's name* *Either FromName or FromCompany must contain a value.* *For customs forms, this element must contain at least two words.*
FromCompany	*Text*	*47*	*Sender's company name* *Either FromName or FromCompany must contain a value.*
ReturnAddress1	Text	47	First delivery address line of sender
ReturnAddress2	*Text*	*47*	*Second delivery address line of sender*
ReturnAddress3	*Text*	*47*	*Third delivery address line of sender.* *Do not use when label type is Domestic or International and a label subtype value is supplied.*
ReturnAddress4	*Text*	*47*	*Fourth delivery address line of sender.* *Do not use when label type is Domestic or International and a label subtype value is supplied.*
FromCity	Text	50	Sender's city Allowed characters: A-Z a-z hyphen period space
FromState	Text	25	Sender's state or province

Element	Type	Size or Values	Description
FromPostalCode	Text	10	Sender's postal code. The format is either ZIP5 only or ZIP+4 for US addresses.
FromZIP4	*Text*	*4*	*+4 add-on for US addresses.* *Ignored if FromPostalCode contains the ZIP4 value.*
FromCountry	*Text*	*50*	*Sender's country. This value should be left blank for USA addresses.*
FromPhone	*Text*	*10*	*Phone number of sender (required for Express Mail and international mail)* *10 digits required (including area code) with no punctuation. Use format: 2125551234*
FromEMail	*Text*	*64*	*E-mail address of sender.*
ToName	*Text*	*47*	*Recipient's name* *For Express Mail and international mail: Either ToName or ToCompany must contain a value.*
ToCompany	*Text*	*47*	*Recipient's company name* *For Express Mail and international mail: Either ToName or ToCompany must contain a value.*
ToAddress1	Text	47	First delivery address line of recipient A value is optional only when ToCompany contains a value. Express Mail labels are limited to four lines in the destination address for all label sizes. 4X5 and 4x4.5 labels are limited to five lines in the destination address.
ToAddress2	*Text*	*47*	*Second delivery address line of recipient*
ToAddress3	*Text*	*47*	*Third delivery address line of recipient.* *Do not use with label type of Domestic.*

Element	Type	Size or Values	Description
ToAddress4	*Text*	*47*	*Fourth delivery address line of recipient.* *Do not use with label type of Domestic.*
ToCity	Text	50	Recipient's city For Domestic Mail, allowed characters are: A-Z a-z hyphen period space
ToState	Text	25	Recipient's state or province. Required for Domestic Mail, optional for International Mail.
ToPostalCode	Text	5 15	Recipient's postal code. For Domestic Mail, the format is ZIP5 (required). For International Mail (optional).
ToZIP4	*Text*	*4*	*+4 add-on for US addresses.*
ToDeliveryPoint	*Text*	*2*	*+2 Delivery Point for US addresses.*
ToCountry	*Text*	*50*	*Recipient's country.* *Refer to Country and Locality Listing, for a table of valid country names.* *Required for International Mail.* *Ignored when label type is International and a label subtype value is supplied. In this case, the ToCountryCode element must be used.*
ToCountryCode	*Text*	*2*	*Two-character country code of the recipient's country.* *Refer to Country and Locality Listing, for a table of valid country names.* *Required when label type is International and a label subtype value is supplied.*

Element	Type	Size or Values	Description
ToPhone	*Text*	*10* *30*	*Recipient's Phone Number.* *For Domestic mail: 10 digits including area code with no punctuation. For example: 2125551234. If supplied value is not in the correct format, it will be ignored.* *For International mail, up to 30 digits with no punctuation.*
ToEMail	*Text*	*64*	*E-mail address of recipient.*
CustomsInfo	*Node*		*Customs Declaration* *Required when using Integrated labels.*
ContentsType	*Text*	*Documents* *Gift* *Merchandise* *Other* *ReturnedGoods* *Sample*	*Category of the customs items.* *Value of ReturnedGoods is not available for labels using Form 2976.*
ContentsExplanation	*Text*		*Explanation of the customs items.* *Required if ContentsType is Other.*
RestrictionType	*Text*	*None* *Other* *Quarantine* *SanitaryPhytosanitaryInspection*	*Default*
RestrictionCommments	*Text*	*25*	*Required if RestrictionType is not None.*
SendersCustomsReference	*Text*	*14*	*Sender's customs reference,*
ImportersCustomsReference	*Text*	*40*	*Importers customs reference.*
LicenseNumber	*Text*	*16*	*License number.*
CertificateNumber	*Text*	*12*	*Certificate number.*
InvoiceNumber	*Text*	*15*	*Invoice number.*
NonDeliveryOption	*Text*	*Abandon* *Return*	*Non-delivery instructions.* *Treat as abandoned.* *Return to sender. (Default)*
InsuredNumber	*Text*	*13*	*For future use.*

Element	Type	Size or Values	Description
EelPfc	*Text*	*35*	*Exemption or Exclusion Legend (EEL) or a Proof of Filing Citation (PFC).Required for shipments to an international destination or to an overseas U.S. Territory.* *It is recommended to supply a value for this element as the USPS will likely require it in the near future.*
CustomsItems	Node		*Must contain at least 1 but no more than 30 CustomsItem nodes.*
CustomsItem		*[1 .. 30]*	
Description	Text	50	Description of the customs item.
Quantity	Numeric	3	Quantity of the customs item. Must be greater than zero.
Weight	Numeric	4	Weight of the customs item. Must be specified in whole ounces and greater than zero and cannot exceed 1120 ounces (70 pounds).
Value	Currency	5.2	Value of the customs item. Must be greater than zero.
HSTariffNumber	*Text*	*6*	*6-digit HS tariff number.*
CountryOfOrigin	*Text*	*2*	*Two character country code of the country where the customs items originated.* *Refer to the Country Code Table for a list of valid country codes.*
The following five sets of customs item elements are ignored when CustomsInfo is supplied.			
CustomsDescription1	Text	50	Description of the first item in the customs declaration. If a value is supplied, then the remaining elements for the first item in the customs declaration must also contain a value. If a value is not supplied, then the remaining elements for the first item in the customs declaration are ignored. Required when requesting an International Mail label or a customs form.

Element	Type	Size or Values	Description
CustomsQuantity1	Numeric	8	Quantity of first item in customs declaration. Required when requesting an International Mail label or a customs form.
CustomsWeight1	Numeric	4	Weight, in ounces, of first item in customs declaration. The value must be specified in whole ounces and greater than zero. Required when requesting an International Mail label or a customs form.
CustomsValue1	Currency	5.2	Value of the first item in the customs declaration. Required when requesting an International Mail label or a customs form.
CustomsCountry1	*Text*	*50*	*Country of Origin for the first item in the customs declaration. This value replaces any value in the OriginCountry tag.* *Refer to Country and Locality Listing, for a table of valid country names.*
CustomsDescription2			*Description of the second item in the customs declaration.* *If a value is supplied, then the remaining elements for the second item in the customs declaration must also contain a value.* *If a value is not supplied, then the remaining elements for the second item in the customs declaration are ignored.*
CustomsQuantity2	*Numeric*	*8*	*Quantity of second item in customs declaration.*
CustomsWeight2	*Numeric*	*4*	*Weight, in ounces, of second item in customs declaration. The value must be specified in whole ounces and greater than zero.*
CustomsValue2	*Currency*	*5.2*	*Value of the second item in the customs declaration.*

Element	Type	Size or Values	Description
CustomsCountry2	*Text*	*50*	*Country of Origin for the second item in the customs declaration.* *Refer to Country and Locality Listing, for a table of valid country names..*
CustomsDescription3	*Text*	*50*	*Description of the third item in the customs declaration.* *If a value is supplied, then the remaining elements for the third item in the customs declaration must also contain a value.* *If a value is not supplied, then the remaining elements for the third item in the customs declaration are ignored.*
CustomsQuantity3	*Numeric*	*8*	*Quantity of third item in customs declaration.*
CustomsWeight3	*Numeric*	*4*	*Weight, in ounces, of third item in customs declaration. The value must be specified in whole ounces and greater than zero.*
CustomsValue3	*Currency*	*5.2*	*Value of the third item in the customs declaration.*
CustomsCountry3	*Text*	*50*	*Country of Origin for the third item in the customs declaration.* *Refer to Country and Locality Listing, for a table of valid country names.*
CustomsDescription4	*Text*	*50*	*Description of the fourth item in the customs declaration.* *If a value is supplied, then the remaining elements for the fourth item in the customs declaration must also contain a value.* *If a value is not supplied, then the remaining elements for the fourth item in the customs declaration are ignored.*
CustomsQuantity4	*Numeric*	*8*	*Quantity of fourth item in customs declaration.*

Element	Type	Size or Values	Description
CustomsWeight4	*Numeric*	*4*	*Weight, in ounces, of fourth item in customs declaration.*
CustomsValue4	*Currency*	*5.2*	*Value of the fourth item in the customs declaration. The value must be specified in whole ounces and greater than zero.*
CustomsCountry4	*Text*	*50*	*Country of Origin for the fourth item in the customs declaration.* *Refer to Country and Locality Listing, for a table of valid country names.*
CustomsDescription5	*Text*	*50*	*Description of the fifth item in the customs declaration.* *If a value is supplied, then the remaining elements for the fifth item in the customs declaration must also contain a value.* *If a value is not supplied, then the remaining elements for the fifth item in the customs declaration are ignored.*
CustomsQuantity5	*Numeric*	*8*	*Quantity of fifth item in customs declaration.*
CustomsWeight5	*Numeric*	*4*	*Weight, in ounces, of fifth item in customs declaration. The value must be specified in whole ounces and greater than zero.*
CustomsValue5	*Currency*	*5.2*	*Value of the fifth item in the customs declaration.*
CustomsCountry5	*Text*	*50*	*Country of Origin for the fifth item in the customs declaration.* *Refer to Country and Locality Listing, for a table of valid country names.*

The following listing shows an example of the LabelRequest XML.

Listing 3-1: LabelRequest XML

```
<LabelRequest Test="YES" LabelType="Default" LabelSize="4X6" ImageFormat="GIF">
  <RequesterID>abcd</RequesterID>
  <AccountID>123456</AccountID>
  <PassPhrase>samplePassPhrase</PassPhrase>
  <MailClass>Priority</MailClass>
```

```
  <DateAdvance>0</DateAdvance>
  <WeightOz>15</WeightOz>
  <MailpieceShape>Parcel</MailpieceShape>
  <Stealth>FALSE</Stealth>
  <Services  InsuredMail="OFF" SignatureConfirmation="ON" />
  <Value>0</Value>
  <Description>Sample Label</Description>
  <PartnerCustomerID>12345ABCD</PartnerCustomerID>
  <PartnerTransactionID>6789EFGH</PartnerTransactionID>
  <ToName>Amine Khechfe</ToName>
  <ToCompany>Endicia</ToCompany>
  <ToAddress1>247 High Street</ToAddress1>
  <ToCity>Palo Alto</ToCity>
  <ToState>CA</ToState>
  <ToPostalCode>84301</ToPostalCode>
  <ToZIP4>0000</ToZIP4>
  <ToDeliveryPoint>00</ToDeliveryPoint>
  <ToPhone>8005763279</ToPhone>
  <FromName>John Doe</FromName>
  <ReturnAddress1>123 Main Street</ReturnAddress1>
  <FromCity>Boise</FromCity>
  <FromState>ID</FromState>
  <FromPostalCode>83702</FromPostalCode>
  <FromZIP4>7261</FromZIP4>
  <FromPhone>8005551212</FromPhone>
  <ResponseOptions PostagePrice="TRUE"/>
</LabelRequest>
```

LabelRequestResponse XML Syntax

The following table describes the XML tags for LabelRequestResponse.

Table 3-2: LabelRequestResponse XML Elements

Element	Type	Size or Values	Description
LabelRequestResponse	**Root**	**-**	**The root node for the XML response.**
Status	Numeric	0 Error code	Indicates if the label request succeeded or failed: Success Failed. The response contains an ErrorMessage element describing the error.
ErrorMessage	*Text*	*150*	*Error message - see Status*
			For EPL2 and ZPLII, the label image contains binary data which should be sent to the Zebra printer without any intermediate conversion.
Base64LabelImage	Base64 string		Image of the label.

Element	Type	Size or Values	Description
			If the Label node is present, then this element will not exist.
Label	*Node*		*Contains one or more <Image> elements for each part of the label.*
Image	Base64 string		First part of the label image.
PartNumber	Attribute, Numeric	1	
Image	*Base64 string*		*Second part of the label image.*
PartNumber	*Attribute, Numeric*	*2*	
:	:	:	:
Image	*Base64 string*		*nth (last) part of the label image.*
PartNumber	*Attribute, Numeric*	*n*	
CustomsForm	*Node*		*Contains one or more <Image> elements for each part of the customs form.*
Image	Base64 string		First part of the customs form image.
PartNumber	Attribute, Numeric	1	
Image	*Base64 string*		*Second part of the customs form image.*
PartNumber	*Attribute, Numeric*	*2*	
:	:	:	:
Image	*Base64 string*		*nth (last) part of the customs form image.*
PartNumber	*Attribute, Numeric*	*n*	
TrackingNumber	Text	Variable Length	For domestic mail, this value is also copied to the PIC element. For international mail and domestic mail using customs forms, this value is also copied to the CustomsNumber element.
PIC	*Text*		*Same as TrackingNumber.* *For domestic mail pieces only.*
CustomsNumber	*Text*	*13*	*Customs form number.*
FinalPostage	Currency	6.3	Total postage and fees for transaction.
TransactionID	Numeric		Unique ID for transaction - corresponds to the piece count for the account used.
TransactionDateTime	Date and Time	YYYYMMDDHHMMSS	Date and time the transaction was created. Expressed in UTC.
PostmarkDate	Date	YYYYMMDD	Indicates the postmark date.

Element	Type	Size or Values	Description
PostageBalance	Currency	6.3	Remaining postage balance after label is generated, in dollars and cents (mils).
PostagePrice	*Node*		*Postage price for transaction.*
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of postage and fees. This value is also copied to the FinalPostage element.*
Postage	*Node*		
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of postage.*
MailService	*Text*	*50*	*Mail Service.*
Zone	*Text*	*0 (local zone), 1 through 8*	*Number of postal rate zones between the sender's and recipient's ZIP Codes.* *Applies only to domestic mail.*
IntraBMC	*Text*	*true* *false*	*Zone is Intra-BMC.* *Zone is Inter-BMC.* *Applies only to domestic mail.*
Pricing	*Text*	*Commercial* *CommercialBase* *CommercialPlus* *NSA* *Retail*	*Commercial pricing.* *Commercial Base pricing.* *Commercial Plus pricing.* *NSA pricing.* *Retail pricing.*
Fees	*Node*		
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of fees.*
CertificateOfMailing	*Currency*	*6.3*	*Certificate of Mailing fee.* *For future use.*
CertifiedMail	*Currency*	*6.3*	*Certified Mail fee.*
CollectOnDelivery	*Currency*	*6.3*	*Collect on Delivery (COD) fee.* *For future use.*
DeliveryConfirmation	*Currency*	*6.3*	*Delivery Confirmation fee.*
ElectronicReturnReceipt	*Currency*	*6.3*	*Electronic Return Receipt fee.*
InsuredMail	*Currency*	*6.3*	*Insured Mail fee.* *Only for USPS insured mailpieces.*
RegisteredMail	*Currency*	*6.3*	*Registered Mail fee.*
RestrictedDelivery	*Currency*	*6.3*	*Restricted Delivery fee.* *For future use.*
ReturnReceipt	*Currency*	*6.3*	*Return Receipt fee.*
ReturnReceiptForMerchandise	*Currency*	*6.3*	*Return Receipt for Merchandise fee.* *For future use.*
SignatureConfirmation	*Currency*	*6.3*	*Signature Confirmation fee.*
SpecialHandling	*Currency*	*6.3*	*Special Handling fee.* *For future use.*

The following listing shows an example of the LabelRequestResponse XML.

Listing 3-2: LabelRequestResponse XML

```
<LabelRequestResponse>
  <Status>0</Status>
  <Base64LabelImage>
    .
    . (label image in base64 string format)
    .
  </Base64LabelImage>
  <PIC>91221234567891234567 83</PIC>
  <TrackingNumber>91221234567891234567 83</TrackingNumber>
  <FinalPostage>6.55</FinalPostage>
  <TransactionID>789</TransactionID>
  <TransactionDateTime>20080509060019</TransactionDateTime>
  <PostmarkDate>20080508</PostmarkDate>
  <PostageBalance>123.450</PostageBalance>
  <PostagePrice TotalAmount="6.55">
   <Postage TotalAmount="4.75">
    <MailService>Priority Mail</MailService>
    <Zone>4</Zone>
    <IntraBMC>FALSE</IntraBMC>
    <Pricing>CommercialBase</Pricing>
   </Postage>
   <Fees TotalAmount="1.8">
    <CertificateOfMailing>0</CertificateOfMailing>
    <CertifiedMail>0</CertifiedMail>
    <CollectOnDelivery>0</CollectOnDelivery>
    <DeliveryConfirmation>0</DeliveryConfirmation>
    <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
    <InsuredMail>0</InsuredMail>
    <RegisteredMail>0</RegisteredMail>
    <RestrictedDelivery>0</RestrictedDelivery>
    <ReturnReceipt>0</ReturnReceipt>
    <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
    <SignatureConfirmation>1.8</SignatureConfirmation>
    <SpecialHandling>0</SpecialHandling>
   </Fees>
  </PostagePrice>
</LabelRequestResponse>
```

4. BUYING POSTAGE API (REQUIRED)

RecreditRequest

To add postage to an account, use the BuyPostage or the BuyPostageXML Web method of the Endicia Label Server Web Service.

The following is a sample HTTP POST request for BuyPostageXML which requires a RecreditRequest XML string as input. The **placeholders** shown need to be replaced with actual values:

```
POST /LabelService/EwsLabelService.asmx/BuyPostageXML HTTP/1.1
Host: www.envmgr.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

recreditRequestXML=<RecreditRequest> ... </RecreditRequest>
```

The response contains a RecreditRequestResponse XML string:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<RecreditRequestResponse> ... </RecreditRequestResponse>
```

RecreditRequest XML Syntax

The following table describes the XML tags for RecreditRequest.

Important notes about all elements:

- Omitting any value causes the Endicia Label Server Web Service to use the default value for that element.
- *Italic* items are optional.

Table 4-1: RecreditRequest XML Elements

Element	Type	Size or Values	Description
RecreditRequest	**Root**	**-**	**The root node for the XML request.**
RequesterID	Text	50	Requester ID (also called Partner ID) is used to uniquely identify the system making the request. This ID is assigned by Endicia.
RequestID	Text	50	Request ID to uniquely identify this Recredit request. This will be returned in response.
CertifiedIntermediary	Node		Certified Intermediary (CI) account authentication information.
AccountID	Numeric	6	Account ID for the Endicia postage account.

Element	Type	Size or Values	Description
PassPhrase	Text	64	Pass Phrase for the Endicia postage account.
RecreditAmount			Amount of postage, in dollars, to add to the account. Either use a predefined amount from the list or enter any value up to 99999.99. Predefined amounts:
	Text	10	Add $10.00 to the account.
		25	Add $25.00 to the account.
		50	Add $50.00 to the account.
		100	Add $100.00 to the account.
		250	Add $250.00 to the account.
		500	Add $500.00 to the account.
		1000	Add $1,000.00 to the account.
		2500	Add $2,500.00 to the account.
		5000	Add $5,000.00 to the account.
		7500	Add $7,500.00 to the account.
		10000	Add $10,000.00 to the account.
		20000	Add $20,000.00 to the account.
	Currency	5.2	Any amount, at least $10.00 and up to $99,999.99, in unit of dollars and rounded to the nearest cent. The minimum amount of postage that can be purchased is $10. The maximum amount is based on the settings of the account.

The following listing shows an example of the RecreditRequest XML.

Listing 4-1: RecreditRequest XML

```
Listing : RecreditRequest XML<RecreditRequest>
  <RequesterID>abcd</RequesterID>
  <RequestID>BP123</RequestID>
  <CertifiedIntermediary>
    <AccountID>123456</AccountID>
    <PassPhrase>samplePassPhrase</PassPhrase>
  </CertifiedIntermediary>
  <RecreditAmount>250</RecreditAmount>
</RecreditRequest>
```

RecreditRequestResponse XML Syntax

The following table describes the XML tags for RecreditRequestResponse.

Table 4-2: RecreditRequestResponse XML Elements

Element	Type	Size or Values	Description
RecreditRequestResponse	**Root**	**-**	**The root node for the XML response.**
Status	Numeric	0 Error code	Indicates if the request to buy postage succeeded or failed: Success Failed. The response contains an ErrorMessage element describing the error.
ErrorMessage	*Text*	*150*	*Error message - see* Status.
RequesterID	Text	50	Return of Requester ID—based on input value.
RequestID	Text	50	Return of Request ID—based on input value.
CertifiedIntermediary	Node		Certified Intermediary (CI) account status.
AccountID	Numeric	6	Account ID for the Endicia postage account.
SerialNumber	Numeric	12	The Serial Number of the last item created on the account.
PostageBalance	Currency	6.3	Remaining postage balance after re-credit in dollars and cents (mils)
AscendingBalance	Currency	6.3	Total amount of postage printed (after re-credit) in dollars and cents (mils)
AccountStatus	Text	A	Account Status flag: Active
DeviceID	Text	12	The account's 12 character Device ID (that appears in the indicium).

The following listing shows an example of a RecreditRequestResponse XML.

Listing 4-2: RecreditRequestResponse XML

```
<RecreditRequestResponse>
  <Status>0</Status>
  <RequesterID>abcd</RequesterID>
  <RequestID>BP123</RequestID>
  <CertifiedIntermediary>
    <AccountID>123456</AccountID>
    <SerialNumber>789</SerialNumber>
    <PostageBalance>76.55</PostageBalance>
    <AscendingBalance>123.45</AscendingBalance>
    <AccountStatus>A</AccountStatus>
    <DeviceID>071V00123456</DeviceID>
  </CertifiedIntermediary>
</RecreditRequestResponse>
```

Please note, the following information about purchasing postage and bank-to-bank postage payment options.

Postage Purchases

- Purchase Methods: Postage purchases are billed when a customer purchases postage.
- Payment Types: Visa, MasterCard, American Express, Discover-Novus, Diner's Club International, Carte Blanche, checks, and direct debits (ACH) from US-based checking accounts, and ACH Credit issued to the USPS Bank account will be accepted. Payment methods are subject to change at any time.
- When an account is initiated, the minimum postage purchase is $10 and the maximum is $500, and a customer's account may contain a maximum of $500 of postage at any given time. An unlimited number of purchases can be made. No sales taxes are applicable to purchases of postage.
  A customer may apply directly with Endicia's support staff (support@endicia.com) to have the postage purchase limits on an account increased after they have used the account and have established a postage buying pattern (usually a couple of weeks).

Endicia Bank-to-Bank Postage Payment Options

- ACH Withdrawal (Direct Debit):
  With the ACH Withdrawal method, you provide information about your checking account, and Endicia will withdraw the funds directly from your checking account and add the amount to your Endicia postage account balance. Endicia uses the USPS-authorized Federal Automated Clearing House (ACH) system to handle all payments. The purchase process in the client works exactly the same way with this method, and you can buy postage at any time to increase the balance in your Endicia account. If this payment method is used, your payment limits can be increased beyond the standard $500 restriction after an initial setup period. The following steps describe how to switch to direct debit:
  1. Log in to your online Endicia account at https://www.endicia.com/Account/LogIn/
  2. Select **Update Profile** > **Payment Method**.

3. Select the **Deduct from my Checking Account…** option.
4. Enter your checking account and routing number.

Alternately, you can also contact support@endicia.com by e-mail to switch.

- ACH Credit (or Push) to the US Postal Service:
  The ACH Credit (Push) method allows you to transfer funds directly from your bank to the USPS. The USPS then transfers those funds to Endicia to add to your account balance. The primary advantages to this method are: no bank account information needs to be provided to Endicia and any amount of postage may purchased. For customers that prefer to purchase in large amounts, this can be quite advantageous. The following steps describe how to use this method.

  1. Instruct your bank to send an ACH TRANSFER (NOT A WIRE TRANSFER) to the USPS "lockbox" at Citibank. The account number is 30564608 and the Citibank ABA routing transit number is 021000089.
  2. Include your Endicia account number on the ACH Credit Transaction. Your bank may have a field for an "Invoice Number", "Order Number", "Individual or Company ID", or a note. If this information is omitted, your Endicia account cannot be credited for ten (10) business days or until another ACH transaction is made with your Endicia Account number in the proper field.
  3. Your bank may require additional verification of the transfer. The vendor name for the transfer is the "US Postal Service (Endicia)" and the tax ID is 41-0760000
  4. Expect the transfer to take 3-5 business to reach your Endicia account.

- Wire transfer to Endicia:
  You may institute a wire transfer of funds from your bank to Endicia's bank account at Union Bank of California, and Endicia will subsequently transfer these funds into the USPS bank account. Please make sure to include your Endicia account number in the wire tranfer notation field. CONTACT US at accounting@endicia.com if you are interested in this method. NOTE: There is a $50 fee for each payment using this method.

- Pay by Check:
  You may mail a check directly to Endicia to purchase postage. Make the check out to "Endicia Internet Postage", add your Endicia account number to the check or check stub, and mail to:

  PSI Systems, Inc.
  Attn: Accounting Department
  247 High St.
  Palo Alto, CA 94301-1041

  The process to credit your account will take five business days after we receive the check.
  Note: There is a $50 fee for each payment using this method.

5. CHANGING THE PASS PHRASE API (REQUIRED)

ChangePassPhraseRequest

To change the Pass Phrase for an account, use the ChangePassPhrase or the ChangePassPhraseXML Web method of the Endicia Label Server Web Service.

The following is a sample HTTP POST request for ChangePassPhraseXML, which requires a ChangePassPhraseRequest XML string as input. The **placeholders** shown need to be replaced with actual values:

```
POST /LabelService/EwsLabelService.asmx/ChangePassPhraseXML HTTP/1.1
Host: www.envmgr.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

changePassPhraseRequestXML=<ChangePassPhraseRequest> ...
</ChangePassPhraseRequest>
```

The response contains a ChangePassPhraseRequestResponse XML string:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<ChangePassPhraseRequestResponse> ... </ChangePassPhraseRequestResponse>
```

ChangePassPhraseRequest XML Syntax

The following table describes the XML tags for ChangePassPhraseRequest.

Important notes about all elements:

- Omitting any value causes the Endicia Label Server Web Service to use the default value for that element.
- *Italic* items are optional.

Table 5-1: ChangePassPhraseRequest XML Elements

Element	Type	Size or Values	Description
ChangePassPhraseRequest	**Root**	**-**	**The root node for the XML request.**
RequesterID	Text	50	Requester ID (also known as Partner ID) is used to uniquely identify the system making the request. This ID is assigned by Endicia.
RequestID	Text	50	Request ID to uniquely identify this Recredit request. This will be returned in response.
CertifiedIntermediary	Node		Certified Intermediary (CI) account authentication information.

Element	Type	Size or Values	Description
AccountID	Numeric	6	Account ID for the Endicia postage account.
PassPhrase	Text	64	Pass Phrase for the Endicia postage account
NewPassPhrase	Text	64	New Pass Phrase for the Endicia postage account. Pass Phrase must be at least 5 characters long with a maximum of 64 characters. For added security, the Pass Phrase should be at least 10 characters long and include more than one word, use at least one uppercase and lowercase letter, one number and one non-text character (e.g. punctuation). A Pass Phrase which has been used previously will be rejected.

The following listing shows an example of the ChangePassPhraseRequest XML.

Listing 5-1: ChangePassPhraseRequest XML

```
<ChangePassPhraseRequest>
  <RequesterID>abcd</RequesterID>
  <RequestID>CPP123</RequestID>
  <CertifiedIntermediary>
    <AccountID>123456</AccountID>
    <PassPhrase>samplePassPhrase</PassPhrase>
  </CertifiedIntermediary>
  <NewPassPhrase>newPassPhrase</NewPassPhrase>
</ChangePassPhraseRequest>
```

ChangePassPhraseRequestResponse XML Syntax

The following table describes the XML tags for ChangePassPhraseRequestResponse.

Table 5-2: ChangePassPhraseRequestResponse XML Elements

Element	Type	Size or Values	Description
ChangePassPhraseRequestResponse	**Root**	**-**	**The root node for the XML response.**
Status	Numeric	0 Error code	Indicates if the request to change Pass Phrase succeeded or failed: Success Failed. The response contains an ErrorMessage element describing the error.
ErrorMessage	*Text*	*150*	*Error message - see Status.*
RequesterID	Text	50	Return of Requester ID—based on input value.
RequestID	Text	50	Return of Request ID—based on input value.

The following listing shows an example of the ChangePassPhraseRequestResponse XML.

Listing 5-2: ChangePassPhraseRequestResponse XML

```
<ChangePassPhraseRequestResponse>
  <Status>0</Status>
  <RequesterID>abcd</RequesterID>
  <RequestID>CPP123</RequestID>
</ChangePassPhraseRequestResponse>
```

6. CALCULATING POSTAGE AND FEES API (SINGLE)

PostageRateRequest

To calculate the postage and fees for a single mailpiece, use the CalculatePostageRate or the CalculatePostageRateXML Web method of the Endicia Label Server Web Service.

The following is a sample HTTP POST request for CalculatePostageRateXML, which requires a PostageRateRequest XML string as input. You must replace the **placeholders** with actual values.

```
POST /LabelService/EwsLabelService.asmx/CalculatePostageRateXML HTTP/1.1
Host: www.envmgr.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

postageRateRequestXML=<PostageRateRequest> ... </PostageRateRequest>
```

The response contains a PostageRateResponse XML string:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<PostageRateResponse> ... </PostageRateResponse>
```

PostageRateRequest XML Syntax

The following table describes the XML tags for PostageRateRequest.

Important notes about all elements:

- Omitting any value causes the Endicia Label Server Web Service to use the default value for that element.
- *Italic* items are optional.

Table 6-1: PostageRateRequest XML Elements

Element	Type	Size or Values	Description
PostageRateRequest	**Root**	**-**	**The root node for the XML request.**
RequesterID	Text	50	Requester ID (also called Partner ID) is used to uniquely identify the system making the request. This ID is assigned by Endicia.
CertifiedIntermediary	Node		Certified Intermediary (CI) account authentication information.
AccountID	Numeric	6	Account ID for the Endicia postage account.
PassPhrase	Text	64	Pass Phrase for the Endicia

Element	Type	Size or Values	Description
			postage account.
MailClass	Text	Express First LibraryMail MediaMail ParcelPost ParcelSelect Priority StandardMail ExpressMailInternational FirstClassMailInternational PriorityMailInternational	Domestic: Express Mail First-Class Mail Library Mail Media Mail Parcel Post Parcel Select Priority Mail Standard Mail International: Express Mail International First-Class Mail International Priority Mail International
Pricing	*Text*	*CommercialBase* *CommercialPlus* *Retail*	*Commercial Base pricing.* *Commercial Plus pricing.* *Retail pricing.* *If this element is not supplied, pricing will be based on the MailClass and any qualified discounts available to AccountID.* **Important!** If this element is set to CommercialPlus, but the account does not qualify for such pricing, the Web method will return an error.
WeightOz	Numeric	4.1	Weight of the package, in ounces.
MailpieceShape	*Text*	*Card* *Letter* *Flat* *Parcel* *LargeParcel* *IrregularParcel* *OversizedParcel* *FlatRateEnvelope* *FlatRatePaddedEnvelope* *SmallFlatRateBox* *MediumFlatRateBox* *LargeFlatRateBox*	*Shape of the mailpiece.* *See the Mailpiece Shape Chart for more information.* *Starting in May 2009, FlatRateBox has been replaced with MediumFlatRateBox.* *The Flat Rate Padded Envelope is only available to qualifying Priority Mail Commercial Plus customers. The Label Server will return an error if this shape is used with any other pricing.*
MailpieceDimensions	*Node*		*Dimensions of the mailpiece.* *Required for calculating Cubic pricing and for Priority Mail mailpieces going to Zones 5-8 which are over 1 cubic foot.* *Ignored if MailpieceShape is set*

Element	Type	Size or Values	Description
			to either LargeParcel or OversizedParcel. *All values must be in inches.*
Length	Numeric	3.3	Length of the mailpiece.
Width	Numeric	3.3	Width of the mailpiece.
Height	Numeric	3.3	Height (or thickness) of the mailpiece.
AutomationRate	*Text*	*TRUE* *FALSE*	*Use applicable automation rate for selected mail class.* *Use retail price. (Default)* *Available only for letter shape mailpiece using First-Class.*
Machinable	*Text*	*TRUE* *FALSE*	*Package is machinable. (Default)* *Package is non-machinable.* *If a Parcel Select mailpiece marked as machinable is over 35 lbs. in weight or its MailpieceShape is set to OversizedParcel, it will automatically be charged the non-machinable rate.*
ServiceLevel	*Text*	*NextDay2ndDayPOToAddressee*	*Next Day/Second Day Post Office to Addressee Service* *Applies only to Express Mail. See Miscellaneous Notes & Features for more information.*
SundayHolidayDelivery	*Text*	 *TRUE* *FALSE*	*For Express Mail only:* *Request Sunday/Holiday Delivery Service.* *Do not deliver on Sunday or holiday. (Default)*
SortType	*Text*	*BMC* *FiveDigit* *MixedBMC* *Nonpresorted* *Presorted* *SCF* *SinglePiece* *ThreeDigit*	*Sort level for applicable mail classes.* *Required for Parcel Select and Standard Mail.* *Defaults to SinglePiece for mailpieces which do not require a sort type.*
Services	*Node*		*Special Services requested for the package.*
CertifiedMail	*Attribute*	*OFF* *ON*	*Certified Mail not requested.* *Certified Mail requested.*
COD	*Attribute*	*OFF* *ON*	*COD not requested.* *COD requested.*

Element	Type	Size or Values	Description
DeliveryConfirmation	*Attribute*	*OFF* *ON*	*Delivery Confirmation not requested. (Default)* *Delivery Confirmation requested.*
ElectronicReturnReceipt	*Attribute*	*OFF* *ON*	*Electronic Return Receipt not requested. (Default)* *Electronic Return Receipt requested.*
InsuredMail	*Attribute*	*OFF* *ON* *UspsOnline* *ENDICIA*	*No insurance requested. (Default)* *USPS Insurance requested.* *USPS Online Insurance requested.* *Endicia Insurance requested (Maximum insurable value: $10,000)* *USPS Online Insurance is available only for mailpieces with Delivery or Signature Confirmation.* *USPS Insurance is not allowed for International Mail.* *Endicia insurance fee is not included in the postage price. It is billed to your account.*
RestrictedDelivery	*Attribute*	*OFF* *ON*	*Restricted Delivery not requested. (Default)* *Restricted Delivery requested.*
ReturnReceipt	*Attribute*	*OFF* *ON*	*Return Receipt not requested. (Default)* *Return Receipt requested.*
SignatureConfirmation	*Attribute*	*OFF* *ON*	*Signature Confirmation not requested. (Default)* *Signature Confirmation requested.*
Value	*Currency*	*5.2*	*Value of the item(s) in the mailpiece*
CODAmount	*Currency*	*5.2*	*COD amount to collect.* *Required if COD is requested for the mailpiece.*
InsuredValue	*Currency*	*5.2*	*Insured value of the mailpiece.* *Required if insurance or COD is requested for the mailpiece.*
EntryFacility	*Text*		*Postal facility where the mail is entered. The zip code of this facility must be specified in POZipCode.*

Element	Type	Size or Values	Description
			Required for Parcel Select and Standard Mail. *DBMC* *DDU* *DSCF* *OBMC* *Other*	*Destination BMC* *Destination Delivery Unit* *Destination Sectional Center Facility* *Origin BMC* *Other postal facility (Default)*
FromPostalCode	Text	5	Sender's postal code. The format is ZIP5. *For Parcel Select and Standard Mail, the value of this element contains the zip code of the postal facility specified in EntryFacility.*
ToPostalCode	Text	 5 15	Recipient's postal code. For Domestic Mail, the format is ZIP5 (required). For International Mail (optional).
ToCountry	Text	50	Recipient's country. Refer to Country and Locality Listing, for a table of valid country names. Required for International Mail. Use either ToCountry or ToCountryCode.
ToCountryCode	*Text*	*2*	*Two character country code of the recipient's country.* *Refer to the Country Code Table for a list of valid country codes.* Required for International Mail. Use either ToCountry or ToCountryCode.
ShipDate	*Date*	*MM/DD/YYYY*	*Date mailpiece is shipped.* *Required for Express Mail Sunday/Holiday Delivery Service. Ignored for other mail classes.*
ShipTime	*Time*	*HH:MM AM or HH:MM PM*	*Time mailpiece is shipped.* *Applies only to Express Mail Sunday/Holiday Delivery Service. Ignored for other mail classes.* *If this element is not supplied, it defaults to 12:01 AM.*

Element	Type	Size or Values	Description
ResponseOptions	*Node*		*Optional XML elements to include in the PostageRateResponse.*
PostagePrice	*Attribute*	*TRUE* *FALSE*	*TRUE means the response contains the PostagePrice node.*

The following listing shows an example of the PostageRateRequest XML.

Listing 6-1: PostageRateRequest XML

```
<PostageRateRequest>
  <RequesterID>abcd</RequesterID>
  <CertifiedIntermediary>
    <AccountID>123456</AccountID>
    <PassPhrase>samplePassPhrase</PassPhrase>
  </CertifiedIntermediary>
  <MailClass>Priority</MailClass>
  <WeightOz>8.7</WeightOz>
  <MailpieceShape>Parcel</MailpieceShape>
  <Machinable>True</Machinable>
  <Services DeliveryConfirmation="OFF" SignatureConfirmation="ON"/>
  <FromPostalCode>95747</FromPostalCode>
  <ToPostalCode>94301</ToPostalCode>
  <ResponseOptions PostagePrice="TRUE"/>
</PostageRateRequest>
```

PostageRateResponse XML Syntax

The following table describes the XML tags for PostageRateResponse.

Table 6-2: PostageRateResponse XML Elements

Element	Type	Size or Values	Description
PostageRateResponse	**Root**	**-**	**The root node for the XML response.**
Status	Numeric	0 Error code	Indicates if the request to calculate postage succeeded or failed: Success Failed. The response contains an ErrorMessage element describing the error.
ErrorMessage	*Text*	*150*	*Error message - see Status*
RequesterID	Text	50	Return of Requester ID - based on input value.
Zone	*Text*	*0 (local zone), 1 through 8*	*Number of postal rate zones between the sender's and recipient's ZIP Codes.* *Applies only to domestic mail.*
Postage	Node		Postage price for requested mail service. The response will not contain this node if PostagePrice in ResponseOptions is set to TRUE.
MailService	Text	50	Mail Service.
Rate	Currency	6.3	Calculated postage price including all fees.

Element	Type	Size or Values	Description
PostagePrice	*Node*		*Postage price for requested Mail Service.* *The response contains this node only if PostagePrice in ResponseOptions is set to TRUE.*
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of postage and fees.*
Postage	*Node*		
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of postage.*
MailService	*Text*	*50*	*Mail Service.*
Zone	*Text*	*0 (local zone), 1 through 8*	*Number of postal rate zones between the sender's and recipient's ZIP Codes.* *Applies only to domestic mail.*
IntraBMC	*Text*	*true* *false*	*Zone is Intra-BMC.* *Zone is Inter-BMC.* *Applies only to domestic mail.*
Pricing	*Text*	*Commercial* *CommercialBase* *CommercialPlus* *NSA* *Retail*	*Commercial pricing.* *Commercial Base pricing.* *Commercial Plus pricing.* *NSA pricing.* *Retail pricing.*
Fees	*Node*		
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of fees.*
CertificateOfMailing	*Currency*	*6.3*	*Certificate of Mailing fee.* *For future use.*
CertifiedMail	*Currency*	*6.3*	*Certified Mail fee.*
CollectOnDelivery	*Currency*	*6.3*	*Collect on Delivery (COD) fee.* *For future use.*
DeliveryConfirmation	*Currency*	*6.3*	*Delivery Confirmation fee.*
ElectronicReturnReceipt	*Currency*	*6.3*	*Electronic Return Receipt fee.*
InsuredMail	*Currency*	*6.3*	*Insured Mail fee.* *Only for USPS insured mailpieces.*
RegisteredMail	*Currency*	*6.3*	*Registered Mail fee.*
RestrictedDelivery	*Currency*	*6.3*	*Restricted Delivery fee.* *For future use.*
ReturnReceipt	*Currency*	*6.3*	*Return Receipt fee.*
ReturnReceiptForMerchandise	*Currency*	*6.3*	*Return Receipt for Merchandise fee.* *For future use.*
SignatureConfirmation	*Currency*	*6.3*	*Signature Confirmation fee.*
SpecialHandling	*Currency*	*6.3*	*Special Handling fee.* *For future use.*

The following listing shows an example of the PostageRateResponse XML.

Listing 6-2: PostageRateResponse XML

```
<PostageRateResponse>
  <Status>0</Status>
  <Zone>2</Zone>
  <PostagePrice TotalAmount="6.55">
    <Postage TotalAmount="4.75">
      <MailService>Priority Mail</MailService>
      <Zone>2</Zone>
      <IntraBMC>TRUE</IntraBMC>
      <Pricing>CommercialBase</Pricing>
    </Postage>
    <Fees TotalAmount="1.8">
      <CertificateOfMailing>0</CertificateOfMailing>
      <CertifiedMail>0</CertifiedMail>
      <CollectOnDelivery>0</CollectOnDelivery>
      <DeliveryConfirmation>0</DeliveryConfirmation>
      <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
      <InsuredMail>0</InsuredMail>
      <RegisteredMail>0</RegisteredMail>
      <RestrictedDelivery>0</RestrictedDelivery>
      <ReturnReceipt>0</ReturnReceipt>
      <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
      <SignatureConfirmation>1.8</SignatureConfirmation>
      <SpecialHandling>0</SpecialHandling>
    </Fees>
  </PostagePrice>
</PostageRateResponse>
```

7. CALCULATE POSTAGE AND FEES API (MULTIPLE)

PostageRatesRequest

To calculate the postage and fees for multiple mail classes for a mailpiece, use the CalculatePostageRates or the CalculatePostageRatesXML Web method of the Endicia Label Server Web Service.

The following is a sample HTTP POST request for CalculatePostageRatesXML, which requires a PostageRatesRequest XML string as input. You must replace the **placeholders** with actual values.

```
POST /LabelService/EwsLabelService.asmx/CalculatePostageRatesXML HTTP/1.1
Host: www.envmgr.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

postageRatesRequestXML=<PostageRatesRequest> ... </PostageRatesRequest>
```

The response contains a PostageRatesResponse XML string:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<PostageRatesResponse> ... </PostageRatesResponse>
```

PostageRatesRequest XML Syntax

The following table describes the XML tags for PostageRatesRequest.

Important notes about all elements:

- Omitting any value causes the Endicia Label Server Web Service to use the default value for that element.
- *Italic* items are optional.

Table 7-1: PostageRatesRequest XML Elements

Element	Type	Size or Values	Description
PostageRatesRequest	**Root**	**-**	**The root node for the XML request.**
RequesterID	Text	50	Requester ID (also called Partner ID) is used to uniquely identify the system making the request. This ID is assigned by Endicia.
CertifiedIntermediary	Node		Certified Intermediary (CI) account authentication information.
AccountID	Numeric	6	Account ID for the Endicia postage account.

Element	Type	Size or Values	Description
PassPhrase	Text	64	Pass Phrase for the Endicia postage account.
MailClass	Text	Domestic International	Calculate rates for all applicable domestic mail classes. Calculate rates for all applicable international mail classes.
WeightOz	Numeric	4.1	Weight of the package, in ounces.
MailpieceShape	*Text*	*Card* *Letter* *Flat* *Parcel* *LargeParcel* *IrregularParcel* *OversizedParcel* *FlatRateEnvelope* *FlatRatePaddedEnvelope* *SmallFlatRateBox* *MediumFlatRateBox* *LargeFlatRateBox*	*Shape of the mailpiece.* *See the Mailpiece Shape Chart for more information.*
MailpieceDimensions	*Node*		*Dimensions of the mailpiece.* *Required for calculating Cubic pricing and for Priority Mail mailpieces going to Zones 5-8 which are over 1 cubic foot.* *Ignored if MailpieceShape is set to either LargeParcel or OversizedParcel.* *All values must be in inches.*
Length	Numeric	3.3	Length of the mailpiece.
Width	Numeric	3.3	Width of the mailpiece.
Height	Numeric	3.3	Height (or thickness) of the mailpiece.
Machinable	*Text*	*TRUE* *FALSE*	*Package is machinable. (Default)* *Package is non-machinable.*
Services	*Node*		*Special Services requested for the package.*
CertifiedMail	*Attribute*	*OFF*	*Certified Mail not requested. (Default)*

Element	Type	Size or Values	Description
		ON	*Certified Mail requested.*
COD	*Attribute*	*OFF* *ON*	*COD not requested. (Default)* *COD requested.*
DeliveryConfirmation	*Attribute*	*OFF* *ON*	*Delivery Confirmation not requested. (Default)* *Delivery Confirmation requested.*
ElectronicReturnReceipt	*Attribute*	*OFF* *ON*	*Electronic Return Receipt not requested. (Default)* *Electronic Return Receipt requested.*
InsuredMail	*Attribute*	*OFF* *ON* *UspsOnline* *ENDICIA*	*No insurance requested. (Default)* *USPS Insurance requested.* *USPS Online Insurance requested.* *Endicia Insurance requested (Maximum insurable value: $10,000)* *USPS Online Insurance is available only for mailpieces with Delivery or Signature Confirmation.* *USPS Insurance is not allowed for International Mail.* *Endicia insurance fee is not included in the postage price. It is billed to your account.*
RestrictedDelivery	*Attribute*	*OFF* *ON*	*Restricted Delivery not requested. (Default)* *Restricted Delivery requested.*
ReturnReceipt	*Attribute*	*OFF* *ON*	*Return Receipt not requested. (Default)* *Return Receipt requested.*
SignatureConfirmation	*Attribute*	*OFF* *ON*	*Signature Confirmation not requested. (Default)* *Signature Confirmation requested.*
CODAmount	*Currency*	*5.2*	*COD amount to collect.* *Required if COD is requested for the mailpiece.*
InsuredValue	*Currency*	*5.2*	*Insured value of the mailpiece.*

Element	Type	Size or Values	Description
			Required if insurance or COD is requested for the mailpiece.
FromPostalCode	Text	5	Sender's postal code. The format is ZIP5.
ToPostalCode	Text	5 15	Recipient's postal code. For Domestic Mail, the format is ZIP5 (required). For International Mail (optional).
ToCountry	Text	50	Recipient's country. Refer to Country and Locality Listing, for a table of valid country names. Required for International Mail. Use either ToCountry or ToCountryCode.
ToCountryCode	*Text*	*2*	*Two character country code of the recipient's country.* *Refer to the Country Code Table for a list of valid country codes.* Required for International Mail. Use either ToCountry or ToCountryCode.

The following listing shows an example of the PostageRatesRequest XML.

Listing 7-1: PostageRatesRequest XML

```
<PostageRatesRequest>
  <RequesterID>abcd</RequesterID>
  <CertifiedIntermediary>
    <AccountID>123456</AccountID>
    <PassPhrase>samplePassPhrase</PassPhrase>
  </CertifiedIntermediary>
  <MailClass>Domestic</MailClass>
  <WeightOz>8.7</WeightOz>
  <MailpieceShape>Parcel</MailpieceShape>
  <Machinable>True</Machinable>
  <Services DeliveryConfirmation="OFF" SignatureConfirmation="ON"/>
  <FromPostalCode>95747</FromPostalCode>
  <ToPostalCode>94301</ToPostalCode>
</PostageRatesRequest>
```

PostageRatesResponse XML Syntax

The following table describes the XML tags for PostageRatesResponse.

Table 7-2: PostageRatesResponse XML Elements

Element	Type	Size or Values	Description
PostageRatesResponse	**Root**	**-**	**The root node for the XML response.**
Status	Numeric	0 Error code	Indicates if the request to calculate postage succeeded or failed: Success Failed. The response contains an ErrorMessage element describing the error.
ErrorMessage	*Text*	*150*	*Error message - see Status*
RequesterID	Text	50	Return of Requester ID - based on input value.
PostagePrice	*Node*		*Postage price for requested Mail Service.* There may be multiple instances of this node for every applicable Mail Service.
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of postage and fees.*
Postage	*Node*		
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of postage.*
MailService	*Text*	*50*	*Mail Service.*
Zone	*Text*	*0 (local zone), 1 through 8*	*Number of postal rate zones between the sender's and recipient's ZIP Codes.* *Applies only to domestic mail.*
IntraBMC	*Text*	*true* *false*	*Zone is Intra-BMC.* *Zone is Inter-BMC.* *Applies only to domestic mail.*
Pricing	*Text*	*Commercial* *CommercialBase* *CommercialPlus* *NSA* *Retail*	*Commercial pricing.* *Commercial Base pricing.* *Commercial Plus pricing.* *NSA pricing.* *Retail pricing.*
Fees	*Node*		
TotalAmount	*Attribute, Currency*	*6.3*	*Total amount of fees.*
CertificateOfMailing	*Currency*	*6.3*	*Certificate of Mailing fee.* *For future use.*
CertifiedMail	*Currency*	*6.3*	*Certified Mail fee.*
CollectOnDelivery	*Currency*	*6.3*	*Collect on Delivery (COD) fee.*
DeliveryConfirmation	*Currency*	*6.3*	*Delivery Confirmation fee.*

Element	Type	Size or Values	Description
ElectronicReturnReceipt	*Currency*	*6.3*	*Electronic Return Receipt fee.*
InsuredMail	*Currency*	*6.3*	*Insured Mail fee.* *Only for USPS insured mailpieces.*
RegisteredMail	*Currency*	*6.3*	*Registered Mail fee.* *For future use.*
RestrictedDelivery	*Currency*	*6.3*	*Restricted Delivery fee.*
ReturnReceipt	*Currency*	*6.3*	*Return Receipt fee.*
ReturnReceiptForMerchandise	*Currency*	*6.3*	*Return Receipt for Merchandise fee.* *For future use.*
SignatureConfirmation	*Currency*	*6.3*	*Signature Confirmation fee.*
SpecialHandling	*Currency*	*6.3*	*Special Handling fee.* *For future use.*

The following listing shows an example of the PostageRatesResponse XML.

Listing 7-2: PostageRatesResponse XML

```
<PostageRatesResponse>
 <Status>0</Status>
 <PostagePrice TotalAmount="4.53">
  <Postage TotalAmount="2.58">
   <MailService>First-Class Mail</MailService>
   <Zone>2</Zone>
   <IntraBMC>true</IntraBMC>
   <Pricing>Retail</Pricing>
  </Postage>
  <Fees TotalAmount="1.95">
   <CertificateOfMailing>0</CertificateOfMailing>
   <CertifiedMail>0</CertifiedMail>
   <CollectOnDelivery>0</CollectOnDelivery>
   <DeliveryConfirmation>0</DeliveryConfirmation>
   <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
   <InsuredMail>0</InsuredMail>
   <RegisteredMail>0</RegisteredMail>
   <RestrictedDelivery>0</RestrictedDelivery>
   <ReturnReceipt>0</ReturnReceipt>
   <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
   <SignatureConfirmation>1.95</SignatureConfirmation>
   <SpecialHandling>0</SpecialHandling>
  </Fees>
 </PostagePrice>
 <PostagePrice TotalAmount="6.75">
  <Postage TotalAmount="4.8">
   <MailService>Priority Mail</MailService>
   <Zone>2</Zone>
   <IntraBMC>true</IntraBMC>
   <Pricing>CommercialBase</Pricing>
```

```
  </Postage>
  <Fees TotalAmount="1.95">
    <CertificateOfMailing>0</CertificateOfMailing>
    <CertifiedMail>0</CertifiedMail>
    <CollectOnDelivery>0</CollectOnDelivery>
    <DeliveryConfirmation>0</DeliveryConfirmation>
    <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
    <InsuredMail>0</InsuredMail>
    <RegisteredMail>0</RegisteredMail>
    <RestrictedDelivery>0</RestrictedDelivery>
    <ReturnReceipt>0</ReturnReceipt>
    <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
    <SignatureConfirmation>1.95</SignatureConfirmation>
    <SpecialHandling>0</SpecialHandling>
  </Fees>
</PostagePrice>
<PostagePrice TotalAmount="14.89">
  <Postage TotalAmount="14.89">
    <MailService>Express Mail</MailService>
    <Zone>2</Zone>
    <IntraBMC>true</IntraBMC>
    <Pricing>CommercialBase</Pricing>
  </Postage>
  <Fees TotalAmount="0">
    <CertificateOfMailing>0</CertificateOfMailing>
    <CertifiedMail>0</CertifiedMail>
    <CollectOnDelivery>0</CollectOnDelivery>
    <DeliveryConfirmation>0</DeliveryConfirmation>
    <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
    <InsuredMail>0</InsuredMail>
    <RegisteredMail>0</RegisteredMail>
    <RestrictedDelivery>0</RestrictedDelivery>
    <ReturnReceipt>0</ReturnReceipt>
    <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
    <SignatureConfirmation>0</SignatureConfirmation>
    <SpecialHandling>0</SpecialHandling>
  </Fees>
</PostagePrice>
<PostagePrice TotalAmount="4.21">
  <Postage TotalAmount="2.26">
    <MailService>Library Mail</MailService>
    <Zone>2</Zone>
    <IntraBMC>true</IntraBMC>
    <Pricing>Retail</Pricing>
  </Postage>
  <Fees TotalAmount="1.95">
    <CertificateOfMailing>0</CertificateOfMailing>
    <CertifiedMail>0</CertifiedMail>
    <CollectOnDelivery>0</CollectOnDelivery>
    <DeliveryConfirmation>0</DeliveryConfirmation>
    <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
    <InsuredMail>0</InsuredMail>
```

```
    <RegisteredMail>0</RegisteredMail>
    <RestrictedDelivery>0</RestrictedDelivery>
    <ReturnReceipt>0</ReturnReceipt>
    <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
    <SignatureConfirmation>1.95</SignatureConfirmation>
    <SpecialHandling>0</SpecialHandling>
  </Fees>
</PostagePrice>
<PostagePrice TotalAmount="4.33">
  <Postage TotalAmount="2.38">
    <MailService>Media Mail</MailService>
    <Zone>2</Zone>
    <IntraBMC>true</IntraBMC>
    <Pricing>Retail</Pricing>
  </Postage>
  <Fees TotalAmount="1.95">
    <CertificateOfMailing>0</CertificateOfMailing>
    <CertifiedMail>0</CertifiedMail>
    <CollectOnDelivery>0</CollectOnDelivery>
    <DeliveryConfirmation>0</DeliveryConfirmation>
    <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
    <InsuredMail>0</InsuredMail>
    <RegisteredMail>0</RegisteredMail>
    <RestrictedDelivery>0</RestrictedDelivery>
    <ReturnReceipt>0</ReturnReceipt>
    <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
    <SignatureConfirmation>1.95</SignatureConfirmation>
    <SpecialHandling>0</SpecialHandling>
  </Fees>
</PostagePrice>
<PostagePrice TotalAmount="6.85">
  <Postage TotalAmount="4.9">
    <MailService>Parcel Post</MailService>
    <Zone>2</Zone>
    <IntraBMC>true</IntraBMC>
    <Pricing>Retail</Pricing>
  </Postage>
  <Fees TotalAmount="1.95">
    <CertificateOfMailing>0</CertificateOfMailing>
    <CertifiedMail>0</CertifiedMail>
    <CollectOnDelivery>0</CollectOnDelivery>
    <DeliveryConfirmation>0</DeliveryConfirmation>
    <ElectronicReturnReceipt>0</ElectronicReturnReceipt>
    <InsuredMail>0</InsuredMail>
    <RegisteredMail>0</RegisteredMail>
    <RestrictedDelivery>0</RestrictedDelivery>
    <ReturnReceipt>0</ReturnReceipt>
    <ReturnReceiptForMerchandise>0</ReturnReceiptForMerchandise>
    <SignatureConfirmation>1.95</SignatureConfirmation>
    <SpecialHandling>0</SpecialHandling>
  </Fees>
</PostagePrice>
```

```
</PostageRatesResponse>
```

8. GETTING ACCOUNT STATUS API

AccountStatusRequest

To get the status of an account, use the GetAccountStatus or the GetAccountStatusXML Web method of the Endicia Label Server Web Service. For accounts with an Active status, the response contains other useful information, such as the remaining postage balance in the account.

The following is a sample HTTP POST request for GetAccountStatusXML which requires an AccountStatusRequest XML string as input. Replace the **placeholders** with actual values:

```
POST /LabelService/EwsLabelService.asmx/GetAccountStatusXML HTTP/1.1
Host: www.envmgr.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

accountStatusRequestXML=<AccountStatusRequest> ... </AccountStatusRequest>
```

The response contains an AccountStatusResponse XML string:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<AccountStatusResponse> ... </AccountStatusResponse>
```

AccountStatusRequest XML Syntax

The following table describes the XML tags for AccountStatusRequest.

Important notes about all elements:

- Omitting any value causes the Endicia Label Server Web Service to use the default value for that element.
- *Italic* items are optional.

Table 8-1: AccountStatusRequest XML Elements

Element	Type	Size or Values	Description
AccountStatusRequest	**Root**	**-**	**The root node for the XML request.**
RequesterID	Text	50	Requester ID (also called Partner ID) is used to uniquely identify the system making the request. This ID is assigned by Endicia.
RequestID	Text	50	Request ID to uniquely identify this Account Status request. This will be returned in response.
CertifiedIntermediary	Node		Certified Intermediary (CI) account authentication information.
AccountID	Numeric	6	Account ID for the Endicia postage account.
PassPhrase	Text	64	Pass Phrase for the Endicia postage account.

The following listing shows an example of the AccountStatusRequest XML.

Listing 8-1: AccountStatusRequest XML

```
<AccountStatusRequest>
  <RequesterID>abcd</RequesterID>
  <RequestID>GAS12345</RequestID>
  <CertifiedIntermediary>
    <AccountID>123456</AccountID>
    <PassPhrase>samplePassPhrase</PassPhrase>
  </CertifiedIntermediary>
</AccountStatusRequest>
```

AccountStatusResponse XML Syntax

The following table describes the XML tags for AccountStatusResponse.

Table 8-2: AccountStatusResponse XML Elements

Element	Type	Size or Values	Description
AccountStatusResponse	**Root**	**-**	**The root node for the XML response.**
Status	Numeric	0 Error code	Indicates if the account status request succeeded or failed: Success Failed. The response contains an ErrorMessage element describing the error.
ErrorMessage	*Text*	*150*	*Error message - see Status*
RequesterID	*Text*	*50*	*Return of Requester ID - based on input value.*
RequestID	*Text*	*50*	*Return of Request ID - based on input value.*

Element	Type	Size or Values	Description
CertifiedIntermediary	*Node*		*Certified Intermediary (CI) account status.* *This node will only be present for an account with an Active status.*
AccountID	*Numeric*	*6*	*Account ID for the Endicia postage account.*
SerialNumber	*Numeric*	*12*	*The Serial Number of the last item created on the account.*
PostageBalance	*Currency*	*6.3*	*Remaining postage balance in dollars and cents (mils).*
AscendingBalance	*Currency*	*6.3*	*Total amount of postage printed in dollars and cents (mils).*
AccountStatus	*Text*	*A*	*Account Status flag:* *Active*
DeviceID	*Text*	*12*	*The account's 12 character Device ID (that appears in the indicium).*

The following listing shows an example of an AccountStatusResponse XML.

Listing 8-2: AccountStatusResponse XML

```
<AccountStatusResponse>
  <Status>0</Status>
  <RequesterID>abcd</RequesterID>
  <RequestID>GAS12345</RequestID>
  <CertifiedIntermediary>
   <AccountID>123456</AccountID>
   <SerialNumber>789</SerialNumber>
   <PostageBalance>76.55</PostageBalance>
   <AscendingBalance>123.45</AscendingBalance>
   <AccountStatus>A</AccountStatus>
   <DeviceID>071V00123456</DeviceID>
  </CertifiedIntermediary>
</AccountStatusResponse>
```

9. REQUESTING A REFUND API

RefundRequest

If you are developing software for use by multiple customers and thus will be maintaining multiple accounts, this chapter demonstrates an automated Web Service that allows your customers to submit refund requests more efficiently. You can collect all the needed information and just submit it through this API. In return, you will receive an acknowledgement message.

Important! The URL for this service is distinct and different than the Endicia Label Server Web Service URL. You will receive this unique URL for this API with the "Welcome" e-mail message.

RefundRequest XML Syntax

The following table describes the XML tags for RefundRequest.

Important notes about all elements:

- Omitting any value causes the Refund Web Service to use the default value for that element.
- *Italic* items are optional.
- The RefundList Node can have many Sub Nodes to do a list of shipments that need refunding.
- As a rule a tracking number is the PICNumber for a shipment. The technical term will always be PICNumber, but it is commonly referred to as a tracking number.
- You do not need both the PICNumber and PieceID for a refund, only one or the other. They are both in the API to give a choice as to which to use.
- You only need to submit one PieceID or one PICNumber per shipment each time you want to request a refund.
- It is recommended that you submit no more than 100 requests in a single XML transaction. If you submit more than 100 requests in a single XML transaction, your request may time out.
- If you include your Pass Phrase in the request, you do not need to submit your Web Password.

Table 9-1: RefundRequest XML Elements

Element	Type	Size or Values	Description
RefundRequest	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.

Element	Type	Size or Values	Description
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
Test	*Text*	*Y or N*	*Is this a test request?*
RefundList	**Node**	**-**	**Container Node For the Refund List**
PICNumber	*Numeric*	*30*	*Package PIC Number (Tracking Number)*
CustomsID	*Text*	*25*	*Package Customs ID (GPM and GAPP tracked shipments only)*
PieceID	*Numeric*	*10*	*Package Piece ID*

The following listing shows an example of the RefundRequest XML for the Refund Request API.

Listing 9-1: RefundRequest XML

```
<?xml version="1.0" encoding="utf-8"?>
<RefundRequest>
        <AccountID></AccountID>
        <WebPassword></WebPassword>
        <Test></Test>
        <RefundList>
                <PICNumber></PICNumber>
                <PieceID></PieceID>
                <CustomsID></CustomsID>
        </RefundList>
</ RefundRequest >
```

RefundResponse XML Syntax

The following table describes the XML tags for RefundResponse.

Table 9-2: RefundResponse XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
RefundResponse	**Root**	**-**	**The root node for the XML response.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	*-*	*Account Login Error Message*

Element	Type	Size or Values	Description
FormNumber	*Numeric*	*6*	*Form Number for Refunded Shipments (Only for those approved)*
Test	*Text*	*Y or N*	*Is this a test request?*
RefundList	**Node**	**-**	**Container Node For the Refund List**
PICNumber	*Numeric*	*30*	*Package PIC Number*
IsApproved	*Boolean*		*YES OR NO*
ErrorMsg	*Text*		*If approved or disapproved, this indicates why.*
PieceID	*Numeric*	*10*	*Package Piece ID*
IsApproved	*Boolean*		*YES OR NO*
ErrorMsg	*Text*		*If approved or disapproved, this will say why*
CustomsID	*Text*	*25*	*Package Customs ID (For GPM and GAPP tracked shipments only)*
IsApproved	*Boolean*		*YES OR NO*
ErrorMsg	*Text*		*If approved or disapproved, this will say why*

The following listing shows an example of the RefundResponse XML (XMLInput Parameter).

Listing 9-2: RefundResponse XML (XMLInput Parameter)

```
<?xml version="1.0" encoding="UTF-8"?>
<RefundResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <FormNumber></FormNumber>
        <Test></Test>
        <RefundList>
                <PICNumber>
                        <IsAppoved></IsApproved>
                        <ErrorMsg></ErrorMsg>
                </PICNumber>
                <PieceID>
                        <IsAppoved></IsApproved>
                        <ErrorMsg></ErrorMsg>
                </PieceID>
                <CustomsID>
                        <IsApproved></IsApproved>
                        <ErrorMsg></ErrorMsg>
                </CustomsID>
        </RefundList>
</RefundResponse>
```

Using form submission: To use this API with a form submission, you must declare both the method and the XML's parameter name in the submission.

Example:

```
<form name="RefundRequest"
      method="POST"
      action="URL TO API&method=RefundRequest">
      <input type="hidden" name="XMLInput" value="XML (Formatted as above)">
</form>
```

OR (through cURL in PHP):

```
$xml = XML (Formatted as above);
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, 'URL TO API');
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS,
'&method=RefundRequest&XMLInput=' . $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');
$soapclient = new soapclient('URL TO API');
```
$xml = *XML (Formatted as above)*;

```
$params = array(
        'XMLInput' => $xml
                );
$soapclient->call('RefundRequest',$params);
echo $soapclient->response;
```

RefundRequest Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
Denied - Must be within 10 days.	Means that the refund was not requested within 10 days of the postage purchase. For electronic refunds this must always be within that 10-day period. (USPS rule)
Denied - Endicia Printable Postage is not neligible for refund.	You should never get this error, it is added in for completeness. We have another product Endicia Printable Postage that cannot ever be refunded.
Denied – Invalid	Given if the ID sent was empty.

Error Code String	Error Message
Denied - Refund already submitted	Given if there already has been a refund request on this item.
Denied - Not Found	Given if there isn't any transaction in our system with that particular ID.
Approved - Less than 10 days.	All is well, Your refund has been submitted and approved.

10. SIGNING UP FOR AN ACCOUNT API

UserSignupRequest

If you are developing software for use by multiple customers and thus will be maintaining multiple accounts, this chapter demonstrates an automated Web Service that will allow your customers to sign up for accounts more efficiently. You can collect all the needed information and just submit it through this API. In return, you will get an acknowledgement message back. The actual account number will still be returned through a separate e-mail message and that will still need to be associated with the account in a separate step. This is necessary to maintain the security model of creating accounts.

Important! The URL for this service is distinct and different than the Endicia Label Server Web Service URL. You will receive this unique URL for this API with the "Welcome" e-mail message.

UserSignupRequest XML Syntax

The following table describes the XML tags for UserSignupRequest.

Important notes about all elements:

- Omitting any value will cause the Account Signup API to use the default value for that element.
- *Italic* items are optional.
- After signing up, the account number will be sent by e-mail message to the customers e-mail address. It will not be sent in the reply. This is for security reasons. In the event of the account number being lost, it can be found using the confirmation number sent in the reply.

Important notes about certain elements:

- **CompanyName**: The # symbol will be replaced with "No.".
- **PhoneNumber, PhoneNumberExtension, PhysicalZipCode, MailingZipCode, and FaxNumber:** Spaces, dashes and periods will be stripped.

Table 10-1: UserSignupRequest XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
UserSignupRequest	**Root**	**-**	**The root node for the XML request.**
CompanyName	*Text*	*100*	*Name of the Customer's Company*
Test	*Text*	*Y or N*	*Is this a test request?*
FirstName	Text	25	Customer's First Name
MiddleInitial	*Text*	*25*	*Customer's Middle Initial*
LastName	Text	50	Customer's Last Name

Element	Type	Size or Values	Description
Title	*Text*	*50*	*Customer's Contact Title*
EmailAddress	Text	120	Customer's E-mail Address
EmailConfirm	Text	120	Customer's E-mail Address for error checking
PhoneNumber	Numeric	0-9 (xxx-xxx-xxxx)	Customer's Phone Number
PhoneNumberExtension	*Numeric*	*0-9*	*Customer's Phone Number Extention*
ICertify	Text	Y N	I certify that all information furnished is accurate and truthful. I also certify that I have read and understand the United States Postal Service Privacy Act Statement, PC Postage Privacy Principles, and all their contents.
FaxNumber	*Text*	*0-9*	*Customer's Fax Number*
PhysicalAddress	Text	200	Customer's Physical Address
PhysicalCity	Text	100	Customer's Physical City
PhysicalState	Text	2	Customer's Physical State (CA, MN, ND, SD, etc.)
PhysicalZipCode	Text	0-9 (xxxxx or xxxxx-xxxx)	Customer's Physical ZIP Code
MailingAddress	*Text*	*200*	*Customers Mailing Address (if different than physical)*
MailingCity	*Text*	*100*	*Customers Mailing City (if different than physical)*
MailingState	*Text*	*2*	*Customers Mailing State (if different than physical)*

Element	Type	Size or Values	Description
MailingZipCode	*Text*	*0-9 (xxxxx or xxxxx-xxxx)*	*Customers Mailing ZIP Code (if different than physical)*
WebPassword	Text	17	Customers Web Password (Needs to be 5-50 characters) **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customers Account Pass Phrase (Needs to be 5-50 characters) **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
ChallengeQuestion	Text	200	Question to be asked if the password is forgotten (Needs to be at least 5 characters)
ChallengeAnswer	Text	100	The answer to the Challenge Phrase (Needs to be at least 5 characters)
BillingType	*Text*	*TS*	*Type of billing (TS Default for LABELSERVER Product)*
PartnerId	Text	4	The unique ID we give you.
ProductType	Text	LABELSERVER	Type of account you are signing up for
CreditCardNumber	Numerical	0-9	Credit Card Number
CreditCardAddress	Text	200	Address used with credit card
CreditCardCity	Text	100	City used for the credit card
CreditCardState	Text	2	State used with the credit card (CA, MN, ND, SD, etc.)
CreditCardZipCode	Text	0-9 (xxxxx or xxxxx-xxxx)	ZIP Code used with the credit card

Element	Type	Size or Values	Description
CreditCardType	Text	V (Visa) M (MasterCard) A (American Express) B (Carte Blanche) N (Discover/Novis) D (Diners Club)	Credit card value
CreditCardExpMonth	Numerical	01 02 03 04 … 12	Two digit representation of the credit card expiration month
CreditCardExpYear	Numerical	2005 2006 2007 …	Four digit representation of the credit card expiration year
PaymentType	Text	CC ACH	Credit Card or Checking Account (respectively)
CheckingAccountNumber	*Numerical*	*0-15*	*Checking account number (Required if Payment Type is ACH)*
CheckingAccountRoutingNumber	*Numerical*	*0-15*	*Routing Number of checking account (Required if payment type is ACH)*
OverrideEmailCheck	*Text*	*Y or N*	*Removes Duplicate E-mail Address Check.*

The following listing shows a syntax example of the UserSignupRequest XML for the Account Signup API.

Listing 10-1: UserSignupRequest XML (XMLInput Parameter)

```
<?xml version="1.0" encoding="utf-8"?>
<UserSignupRequest>
        <Test></Test>
        <CompanyName></CompanyName>
        <FirstName></FirstName>
        <MiddleInitial></MiddleInitial>
        <LastName></LastName>
        <Title></Title>
        <EmailAddress></EmailAddress>
        <EmailConfirm></EmailConfirm>
        <PhoneNumber></PhoneNumber>
        <PhoneNumberExtension></PhoneNumberExtension>
        <ICertify></ICertify>
        <FaxNumber></FaxNumber>
        <PhysicalAddress></PhysicalAddress>
        <PhysicalCity></PhysicalCity>
        <PhysicalState></PhysicalState>
        <PhysicalZipCode></PhysicalZipCode>
        <MailingAddress></MailingAddress>
```

```
        <MailingCity></MailingCity>
        <MailingState></MailingState>
        <MailingZipCode></MailingZipCode>
        <WebPassword></WebPassword>
        <PassPhrase></PassPhrase>
        <ChallengeQuestion></ChallengeQuestion>
        <ChallengeAnswer></ChallengeAnswer>
        <BillingType></BillingType>
        <PartnerId></PartnerId>
        <ProductType></ProductType>
        <CreditCardNumber></CreditCardNumber>
        <CreditCardAddress></CreditCardAddress>
        <CreditCardCity></CreditCardCity>
        <CreditCardState></CreditCardState>
        <CreditCardZipCode></CreditCardZipCode>
        <CreditCardType></CreditCardType>
        <CreditCardExpMonth></CreditCardExpMonth>
        <CreditCardExpYear></CreditCardExpYear>
        <PaymentType></PaymentType>
        <CheckingAccountNumber></CheckingAccountNumber>
        <CheckingAccountRoutingNumber></CheckingAccountRoutingNumber>
        <OverrideEmailCheck>N</OverrideEmailCheck>
</UserSignupRequest>
```

UserSignupResponse XML Syntax

The following table describes the XML tags for UserSignupResponse.

Table 10-2: UserSignupResponse XML Elements

Element	Type	Size or Values	Description
UserSignupResponse	**Root**	**-**	**The root node for the XML response.**
ErrorMsg	*Text*	*150*	*Error message*
ConfirmationNumber	Numerical	6	Confirmation Number of sign-up
Test	*Text*	*Y or N*	*Is this a test request?*
DateTime	Text	"mm/dd/yyyy – hh:mm:ss GMT"	Date and Time of Signup
ProductType	Text	LABELSERVER	Echoes your product

The following listing shows an example of the UserSignupResponse XML.

Listing 10-2: UserSignupResponse XML

```
<?xml version="1.0" encoding="UTF-8"?>
<UserSignupResponse>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <ConfirmationNumber></ConfirmationNumber>
        <DateTime></DateTime>
        <ProductType></ProductType>
</UserSignupResponse>
```

Table 10-3: UserSignupResponse Error Codes

Error Code String	Error Message
XMLInput Error	XMLInput improperly formatted.
We have detected that you already have an account with this email address	Do you really want another account? Use <OverrideEmailCheck />.
You must include your first name	Error returned for missing node.
You must include your last name	Error returned for missing node.
You must include your email address	Error returned for missing node.
You must include the confirmation of your email address	Error returned for missing node.
You must include your phone number	Error returned for missing node.
You must state that you agree to our terms and conditions	Error returned for missing node.
You must include your physical address	Error returned for missing node.
You must include your physical city	Error returned for missing node.
You must include your physical state	Error returned for missing node.
You must include your physical zip code	Error returned for missing node.
You must include your Web password	Error returned for missing node.
You must include your pass phrase	Error returned for missing node.
You must include your challenge question	Error returned for missing node.
You must include your challenge answer	Error returned for missing node.
You must include the credit card number	Error returned for missing node.
You must include the credit card address	Error returned for missing node.
You must include the credit card city	Error returned for missing node.
You must include the credit card state	Error returned for missing node.
You must include the credit card zip code	Error returned for missing node.
You must include the credit card type	Error returned for missing node.
Physical Address Does Not Exist.	Error returned for invalid data.
Mailing Address Does Not Exist.	Error returned for invalid data.
You Must Agree to the Terms and Conditions	Error returned for invalid data.
You must enter your first name	Error returned for invalid data.
You must enter your physical address	Error returned for invalid data.
You must enter your physical city	Error returned for invalid data.
You must enter your physical state	Error returned for invalid data.

You must enter your physical zip code	Error returned for invalid data.
You have entered an invalid physical zip code, must be an integer	Error returned for invalid data.
You have entered an invalid physical zip code, must be 5 or 9 digits	Error returned for invalid data.
The Internet Password and the Pass Phrase cannot be the same.	Error returned for invalid data.
The Challenge Phrase and the Challenge Answer cannot be the same.	Error returned for invalid data.
Your Internet Password is too small. It must be between 5 and 17 characters.	Error returned for invalid data.
Your Pass Phrase is too small. It must be between 5 and 64 characters.	Error returned for invalid data.
Your Challenge Question is too small. It must be between 5 and 50 characters.	Error returned for invalid data.
Your Challenge Answer is too small. It must be between 5 and 50 characters.	Error returned for invalid data.
Credit card month invalid.	Error returned for invalid billing data.
You must include the credit card expiration month	Error returned for invalid billing data.
Credit card year invalid.	Error returned for invalid billing data.
You must include the credit card expiration year	Error returned for invalid billing data.
You must include the postage payment type	Error returned for invalid billing data.
You must include the product type	Error returned for invalid billing data.
Billing type not found	Error returned for invalid billing data.
Postage payment type not found, must be CC or ACH	Error returned for invalid billing data.
Please select a Credit Card Type.	Error returned for invalid billing data.
Please select a Credit Card Expiration Date.	Error returned for invalid billing data.
Please select a Credit Card Expiration Date.	Error returned for invalid billing data.
Invalid Expiration Year, the year cannot be less than the current year.	Error returned for invalid billing data.
Invalid Expiration Month, the month cannot be less than the current month.	Error returned for invalid billing data.
Card number and card type do not match.	Error returned for invalid billing data.
Card number seems to be missing digits.	Error returned for invalid billing data.
Does not appear to be a supported card.	Error returned for invalid billing data.
Credit card number does not appear valid.	Error returned for invalid billing data.
Credit Card Number Can Only Contain Digits 0-9	Error returned for invalid billing data.
No credit card number entered.	Error returned for invalid billing data.
If paying by ACH you must include your checking account number	Error returned for invalid billing data.
Your checking account number seems to be invalid	Error returned for invalid billing data.
If paying by ACH you must include your routing number	Error returned for invalid billing data.
Your checking routing number seems to be invalid	Error returned for invalid billing data.

Using form submission: To use this API with a form submission you must declare both the method and the XML's parameter name in the submission.

Example:

```
<form name="UserSignupRequest"
      method="POST"
      action="URL TO API?method=UserSignup">
      <input type="hidden" name="XMLInput" value="XML (Formatted as above)">
</form>
```

OR (through cURL in PHP):

```
$xml = XML (Formatted as above);
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, 'URL TO API');
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS, '&method=UserSignup&XMLInput='
. $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');

$soapclient = new soapclient('URL TO API');
```

$xml = *XML (Formatted as above)*;

```
$params = array(
        'XMLInput' => $xml
                );

$soapclient->call('UserSignup',$params);

echo $soapclient->response;
```

11. REQUESTING PACKAGE STATUS API

StatusRequest

If you are developing software for use by multiple customers and thus will be maintaining multiple accounts, this chapter demonstrates an automated Web Service that allows your customers to submit status requests more efficiently. You can collect all the needed information and just submit it through this API. In return, you will get an acknowledgement message back.

Important! The URL for this service is distinct and different than the Endicia Label Server Web Service URL. You will receive this unique URL for this API with the "Welcome" e-mail message.

StatusRequest XML Syntax

The following table describes the XML tags for StatusRequest.

Important notes about all elements:

- Omitting any value will cause the Status Web Service to use the default value for that element.
- *Italic* items are optional.
- The StatusList Node can have many sub-nodes to do a list of shipments that need searching.
- As a rule a tracking number is the PICNumber for a shipment. The technical term will always be PICNumber, but it is commonly referred to as a tracking number.
- You do not need both the PICNumber and PieceID for a status request, just one or the other. They are both in the API to give a choice as to which to use.
- You only need to submit one PieceID or one PICNumber per shipment each time you want to request a status.
- The API is limited to 100 items per request.
- If you include your Pass Phrase in the request, you do not need to submit your Web password.

Table 11-1: StatusRequest XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
StatusRequest	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
Test	*Text*	*Y or N*	*Is this a test request?*
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
StatusList	**Node**	**-**	**Container Node For the Status List – Limited to any mixture of 100 of the following node types: PICNumber, CustomsID, and PieceID.**
PICNumber	*Numeric*	*30*	*Package PIC Number (Tracking Number)*
CustomsID	*Text*	*25*	*Package Customs ID (GPM and GAPP tracked shipments only)*
PieceID	*Numeric*	*10*	*Package Piece ID*

The following listing shows a syntax example of the StatusRequest XML for the Refund Request API.

Listing 11-1: StatusRequest XML

```
<?xml version="1.0" encoding="utf-8"?>
<StatusRequest>
        <AccountID></AccountID>
        <Test></Test>
        <WebPassword></WebPassword>
        <StatusList>
                <PICNumber></PICNumber>
                <PieceID></PieceID>
                <CustomsID></CustomsID>
        </StatusList>
</StatusRequest>
```

StatusResponse XML Syntax

The following table describes the XML tags for StatusResponse.

Table 11-2: StatusResponse XML Elements

Element	Type	Size or Values	Description
StatusResponse	**Root**	**-**	**The root node for the XML response.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	*-*	*Account Login Error Message*
Test	*Text*	*Y or N*	*Is this a test request?*
StatusList	**Node**	**-**	**Container Node For the Status List**
PICNumber	*Numeric*	*30*	*Package PIC Number*
Status	*Text*		*Status of the Shipment*
StatusCode	*Text*	*1*	*USPS Status Code of the Shipment*
PieceID	*Numeric*	*10*	*Package Piece ID*
Status	*Text*		*Status of the Shipment*
StatusCode	*Text*	*1*	*USPS Status Code of the Shipment*
CustomsID	*Text*	*25*	*Package Customs ID (For GPM and GAPP tracked shipments only)*
Status	*Text*		*Status of the Shipment*
StatusCode	*Text*	*1*	*USPS Status Code of the Shipment*

The following listing shows an example of the StatusResponse XML.

Listing 11-2: StatusResponse XML

```
<?xml version="1.0" encoding="UTF-8"?>
<StatusResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <StatusList>
                <PICNumber>
                        <Status></Status>
                        <StatusCode></StatusCode>
                </PICNumber>
                <PieceID>
                        <Status></Status>
                        <StatusCode></StatusCode>
                </PieceID>
                <CustomsID>
                        <Status></Status>
                        <StatusCode></StatusCode>
                </CustomsID>
        </StatusList>
</StatusResponse>
```

Using form submission: To use this API with a form submission, you must declare both the method and the XML's parameter name in the submission.

Example:

```
<form name="StatusRequest"
      method="POST"
      action="URL TO API&method=StatusRequest">
      <input type="hidden" name="XMLInput" value="XML (Formatted as above)">
</form>
```

OR (through cURL in PHP):

```
$xml = XML (Formatted as above);
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, 'URL TO API');
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS,
'&method=StatusRequest&XMLInput=' . $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');
$soapclient = new soapclient('URL TO API');
```
$xml = *XML (Formatted as above)*;

```
$params = array(
        'XMLInput' => $xml
                );$soapclient->call('StatusRequest',$params);

echo $soapclient->response;
```

StatusRequest Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
Not Found	Happens if the item does not belong to the account making the request, or the item is not in our system at all.
Found – No Status	Happens if the USPS has not updated the status of the item, but it is found in our system.

StatusCode Definitions

Note: These are not exact definitions.

Error Code String	Error Message
-1	Not Found
0	Found – No Status
A	Logged at USPS
I	Scanned in Route
D	Delivered
L	Notice Left for Recipient
F	Forwarded to New Address
R	Returned to Sender
U	Undeliverable
N	New Item
X	Other

12. REQUESTING TRANSACTION LISTS API

GetTransactionsListing

If you are developing software for use by multiple customers and thus will be maintaining multiple accounts, this chapter demonstrates an automated Web Service that allows your customers to get transaction listings more efficiently. You can collect all the needed information and just submit it through this API. In return, you will get an acknowledgement message back.

Important! The URL for this service is distinct and different than the Endicia Label Server Web Service URL. You will receive this unique URL for this API with the "Welcome" e-mail message.

GetTransactionsListing XML Syntax

The following table describes the XML tags for GetTransactionsListing.

Important notes about all elements:

- Omitting any value will cause the Get Transactions Listing Web Service to use the default value for that element.
- *Italic* items are optional.
- The TrackingList Node can have many Sub Nodes to do a list of shipments that need searching.
- As a rule a tracking number is the PICNumber for a shipment. The technical term will always be PICNumber, but it is commonly referred to as a tracking number.
- You do not need both the PICNumber and PieceID for a status request, just one or the other. They are both in the API to give a choice as to which to use.
- You only need to submit one PieceID or one PICNumber per shipment each time you want to request a status. GetTransactionListing returns a maximum of 50 results per request.
- Date ranges are limited to 7 days.
- The maximum number of results is 50.
- If used with no optional inputs, it will return the latest 50 transactions with the latest one being first sorted by time.
- If you include your Pass Phrase in the request, you do not need to submit your Web Password.

Table 12-1: GetTransactionsListing XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
TransactionsListingRequest	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
Test	*Text*	*Y or N*	*Is this a test request?*
TransactionType	*Text*	*Prints* *Purchases* *All*	*Returns print transactions only (default)* *Returns purchase transactions only* *Returns both print and purchase transactions*
CostCenter	*Text*	*All* *XXXX*	*Includes CostCenter in response.* *Returns transactions with specified CostCenter, and includes CostCenter in response*
FromAddress	*Text*	*Y or N*	*Do you want the from address information returned?*
StartDate	*Date*	*Mm/dd/yyyy*	*Start Date of the Query*
StartTime	*Time*	*Hh:mm:ss*	*Start Time of the Query*

Element	Type	Size or Values	Description
EndDate	*Date*	*Mm/dd/yyyy*	*End Date of the Query*
EndTime	*Time*	*Hh:mm:ss*	*End Time of the Query*
ToZipCode	*ZipCode*	*XXXXX*	*Destination ZIP Code To Search For*
Results	*Int*	*1 – 50*	*Maximum Results Returned (If more than 50 is used, it will default to 50)*
TrackingList	***Node***	***-***	***Container Node For the Tracking List***
PICNumber	*Numeric*	*30*	*Package PIC Number (Tracking Number)*
CustomsID	*Text*	*25*	*Package Customs ID (GPM and GAPP tracked shipments only)*
PieceID	*Numeric*	*10*	*Package Piece ID*

The following listing shows a syntax example of the StatusRequest XML for the Refund Request API.

Listing 12-1: GetTransactionsListing XML

```
<?xml version="1.0" encoding="utf-8"?>
<TransactionsListingRequest>
        <AccountID></AccountID>
        <WebPassword></WebPassword>
        <Test></Test>
        <TransactionType></TransactionType>
        <StartDate></StartDate>
        <StartTime></StartTime>
        <EndDate></EndDate>
        <EndTime></EndTime>
        <Results></Results>
        <ToZipCode></ToZipCode>
        <FromAddress></FromAddress>
        <TrackingList>
                <PICNumber></PICNumber>
                <PieceID></PieceID>
                <CustomsID></CustomsID>
        </TrackingList>
</TransactionsListingsRequest>
```

TransactionsListingResponse XML Syntax

The following table describes the XML tags for TransactionsListingResponse.

Table 12-2: GetTransactionsListing Response XML Elements

Element	Type	Size or Values	Description
TransactionsListingResponse	**Root**	**-**	**The root node for the XML response.**
TransactionType	Text		Transaction Type – Print or Purchase
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	-	*Account Login Error Message*
Test	*Text*	*Y or N*	*Is this a test request?*
TransactionResults	**Node**	**-**	**Container Node For the Transaction List**
Transaction	Node	-	Container Node For the Transaction
PieceID	Numeric		Piece Number of the Shipment
Amount	Numeric		Value of the Postage
Weight	Text		Weight of the Shipment
TransactionDateTime	DateTime	Mm/dd/yyyy hh:mm:ss	Date and Time the Shipment was Created

Element	Type	Size or Values	Description
PostmarkDate	Date	Mm/dd/yyyy	Date of the Postmark on the Shipment
MailClass	Text		List of the Shipment Definition Delimited By ';' (For example: "First-Class; Delivery Confirmation")
CustomsID	Text		Customs ID of the Shipment
PICNumber	*Text*		*Tracking Number of the Shipment*
ToName	Text		Name of the Person Receiving the Shipment
ToCompany	Text		Company Name of the Receiving Party
ToAddress	Text		Address of the Receiving Party
ToCity	Text		City of the Receiving Party
ToState	Text		State of the Receiving Party
ToZipCode	Text	XXXXX-XXXX	ZIP Code of the Receiving Party
ToPhone	Text	XXXXXXXXXX	Phone Number of the Receiving Party
ToEmail	Text		e-mail Address of the Receiving Party
ToCountry	Text		Country of the Receiving Party
FromName	*Text*		*Name of the Person Sending the Shipment. Shown if FromAddress is Y.*
FromCompany	*Text*		*Company Name of the Sending Party. Shown if FromAddress is Y.*
FromAddress	*Text*		*Address of the Sending Party. Shown if FromAddress is Y.*
FromCity	*Text*		*City of the Sending Party. Shown if FromAddress is Y.*
FromState	*Text*		*State of the Sending Party. Shown if FromAddress is Y.*
FromZipCode	*Text*	*XXXXX-XXXX*	*ZIP Code of the Sending Party. Shown if FromAddress is Y.*
FromPhone	*Text*	*XXXXXXXXXX*	*Phone Number of the Sending Party. Shown if FromAddress is Y.*
FromEmail	*Text*		*E-mail Address of the Sending Party. Shown if FromAddress is Y.*
DeliveryDateTime	*DateTime*	*Mm/dd/yyyy hh:mm:ss*	*Date and Time of the Delivery*

Element	Type	Size or Values	Description
StatusCode	Text		USPS Status Code of the Shipment
Status	Text		USPS Status of the Shipment
RefundCode	*Char*	*1*	*Refund Code of the Shipment*
RefundStatus	*Text*		*Refund Definition of the Shipment*

The following listing shows an example of the GetTransactionsListing Response XML.

Listing 12-2: GetTransactionsListing Response XML

```
<?xml version="1.0" encoding="UTF-8"?>
<TransactionsListingResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <TransactionResults>
                <Transaction>
                   <TransactionType></TransactionType>
                   <PieceID></PieceID>
                   <Amount></Amount>
                   <Weight></Weight>
                   <TransactionDateTime></TransactionDateTime>
                   <PostmarkDate></PostmarkDate>
                   <MailClass></MailClass>
                   <CustomsID></CustomsID>
                   <PICNumber></PICNumber>
                   <ToName></ToName>
                   <ToCompany></ToCompany>
                   <ToAddress></ToAddress>
                   <ToCity></ToCity>
                   <ToState></ToState>
                   <ToZipCode></ToZipCode>
                   <ToPhone></ToPhone>
                   <ToEmail></ToEmail>
                   <ToCountry></ToCountry>
                   <FromCompany</FromCompany>
                   <FromName></FromName>
                   <FromAddress></FromAddress>
                   <FromCity></FromCity>
                   <FromState></FromState>
                   <FromZipCode></FromZipCode>
                   <FromPhone></FromPhone>
                   <FromEmail></FromEmail>
                   <DeliveryDateTime></DeliveryDateTime>
                   <StatusCode></StatusCode>
                   <Status></Status>
                   <RefundCode></RefundCode>
                   <RefundStatus></RefundStatus>
                </Transaction>
        </TransactionResults>
</TransactionsListingResponse>
```

GetTransactionsListing Response Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
ERROR 104: Please limit requests to seven days	Please limit requests to seven days

GetTransactionsListing Response Definitions

Note: These are not exact definitions.

Error Code String	Error Message
-1	Not Found
0	Found–No Status
A	Logged at USPS
I	Scanned in Route
D	Delivered
L	Notice Left for Recipient
F	Forwarded to New Address
R	Returned to Sender
U	Undeliverable
N	New Item
X	Other

GetTransactionsListing Refund Code and Refund Status

Refund Code	Refund Status
X	Refund Submitted
Y	Refund Validated
R	Refund Rejected
A,C, S, U	Refunded
M	Manual Refund

13. USING THE SCAN FORM API

SCANRequest

If you are developing software for use by multiple customers and thus will be maintaining multiple accounts, this chapter demonstrates an automated Web Service that allows your customers to use the SCAN Service. You can collect all the needed information and just submit it through this API. In return, you will get an acknowledgement message back.

Important! The URL for this service is distinct and different than the Endicia Label Server Web Service URL. You will receive this unique URL for this API with the "Welcome" e-mail message.

SCANRequest XML Syntax

The following table describes the XML tags for SCANRequest

Important notes about all elements:

Omitting any value will cause the SCAN Request Web Service to use the default value for that element.

- *Italic* items are optional.
- The PickupList Node can have many Sub Nodes to do a list of shipments that need to be added to the form.
- As a rule a tracking number is the PICNumber for a shipment. The technical term will always be PICNumber, but it is commonly referred to as a tracking number.
- You do not need PICNumber, CustomsID, and PieceID, only one of them. They are all in the API to give a choice as to which to use.
- You may submit as many shipments as you want per request.
- If you include your Pass Phrase in the request, you do not need to submit your Web Password.

Table 13-1: SCANRequest XML Elements

Element	Type	Size or Values	Description
SCANRequest	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.

Element	Type	Size or Values	Description
Test	*Text*	*Y or N*	*Is this a test transaction?*
FromName	*Text*	*50*	*Name On the form (defaults to the name on the account)*
FromAddress	*Text*	*100*	*Address on the form (defaults to the address on the account)*
FromCity	*Text*	*50*	*City on the form (defaults to the city on the account)*
FromState	*Text*	*2*	*State abbreviation on the form (defaults to the state on the account)*
FromZipCode	*Text*	*5 or 10 digits xxxxx or xxxxx-xxxx*	*ZIP Code on the form (defaults to the zip code on the form)*
SubmissionID	*Text*		*Submission ID of the form (for re-printing a previously created form)*
ImageFormat	*Text*	*PDF, GIF, PNG, or JPG*	*Format of the image returned in base64 format (GIF is default).*
CostCenter	*Numeric*		*Can be used instead of SCANList to auto-add all pieces with that grouping code.*
DPI	*Numeric*	*96, 150, or 300*	*Defaults to 96, but can be 96,150, or 300.*
SCANList	**Node**	**-**	**Container Node For the SCAN List**
PICNumber	*Numeric*	*30*	*Package PIC Number (Tracking Number)*
CustomsID	*Text*	*25*	*Package Customs ID (GPM and GAPP tracked shipments only)*

Element	Type	Size or Values	Description
PieceID	*Numeric*	*10*	*Package Piece ID*

The following listing shows a syntax example of the SCANRequest XML for the SCAN Request API.

Listing 13-1: SCANRequest XML

```
<?xml version="1.0" encoding="utf-8"?>
<SCANRequest>
        <AccountID></AccountID>
        <WebPassword></WebPassword>
        <Test></Test>
        <DPI></DPI>
        <SCANList>
                <PICNumber></PICNumber>
                <PieceID></PieceID>
                <CustomsID></CustomsID>
        </SCANList>
</SCANRequest>
```

SCANResponse XML Syntax

The following table describes the XML tags for SCANResponse.

Table 13-2: SCANResponse XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
SCANResponse	**Root**	**-**	**The root node for the XML response.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	-	*Account Login Error Message*
Test	*Text*	*Y or N*	*Is this a test transaction?*
SubmissionID	Numeric		Submission ID of the SCAN Form
ImageFormat	Text	GIF, PNG, or JPG	Image format of the SCAN Form
SCANForm	Base64 encoded image		SCANForm Image

The following listing shows an example of the SCANResponse XML.

Listing 13-2: SCANResponse XML

```
<?xml version="1.0" encoding="UTF-8"?>
<SCANResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <SubmissionID></SubmissionID>
        <ImageFormat></ImageFormat>
        <DPI></DPI>
        <SCANForm></SCANForm>
</SCANResponse>
```

Using form submission: To use this API with a form submission you must declare both the method and the XML's parameter name in the submission.

Example:

```
<form name="SCANRequest"
      method="POST"
      action="URL TO API&method=SCANRequest">
      <input type="hidden" name="XMLInput" value="XML (Formatted as above)">
</form>
```

OR (through cURL in PHP):

```
$xml = XML (Formatted as above);
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, 'URL TO API');
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS,
'&method=SCANRequest&XMLInput=' . $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');

$soapclient = new soapclient('URL TO API');
```

$xml = *XML (Formatted as above)*;

```
$params = array(
        'XMLInput' => $xml
                );

$soapclient->call('SCANRequest',$params);

echo $soapclient->response;
```

SCANRequest Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
XMLInput Error	XML is not formatted correctly.
Error 102: No shipments added to SCAN. Please check tracking numbers.	Either the shipments have already been added to another form, or they do not qualify for SCAN.
Error 105: SCAN Form Not Found	If the SCAN form cannot be found for that account.

14. REQUESTING A CARRIER PICKUP API'S

If you are developing software for use by multiple customers and thus will be maintaining multiple accounts, this chapter demonstrates an automated Web Service that will allow your customers to submit carrier pickup requests more efficiently. You can collect all the needed information and just submit it through this API. In return, you will receive an acknowledgement message.

The Requesting a Carrier Pickup API includes:

- Scheduling a Carrier Pickup API
- Cancelling a Carrier Pickup Request API
- Changing a Carrier Pickup Request API
- Scheduling a Multiple Location Carrier Pickup Request API

Important! The URL for this service is distinct and different than the Endicia Label Server Web Service URL. You will receive this unique URL for this API with the "Welcome" e-mail message.

Important notes about all elements:

- Omitting any value will cause the Web Service to use the default value for that element.
- *Italic* items are optional.
- The PickupList Node, if needed, can have many Sub Nodes to do a list of shipments that need to be added to the carrier pickup.
- As a rule a tracking number is the PICNumber for a shipment. The technical term will always be PICNumber, but it is commonly referred to as a tracking number.
- You do not need both the PICNumber, CustomsID and PieceID, just one of them. They are all in the API to give a choice as to which to use.
- You may submit as many shipments as you want per request.

Important notes about these API's:

- Express Mail International is counted as "Express".
- Priority Mail International is counted as "Priority".
- At least one Express or Priority mail shipment must be included in the pickup.
- A pickup for the next day must be scheduled prior to 2AM EST the day of the pickup.
- A pickup for the next day can be changed and cancelled prior to 2AM EST the day of the pickup.
- You must have the same address information and confirmation number to change or cancel a pickup.
- **You need special permission to use the Multiple Location Carrier Pickup API.** You must contact us to become approved.
- Locations for the Multiple Location Carrier Pickup API are grouped by the Cost Center tag. (LabelRequest XML)
- Address information for the Multiple Location Carrier Pickup defaults to first shipment found in that Cost Center
- If you include your Pass Phrase in the request, you do not need to submit your Web Password.

14A. SCHEDULING A CARRIER PICKUP API

CarrierPickupRequest

Table 14a-1: Carrier Pickup Request API

Element	Type	Size or Values	Description
CarrierPickupRequest	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
Test	*Text*	*Y or N*	*Is this a test transaction?*
UseAddressOnFile	*Text*	*Y or N*	*Use your physical address we have on file (Default Y)*
FirstName	*Text*	*100*	*First name on the pickup (defaults to the first name on the account)*
LastName	*Text*	*100*	*Last name on the pickup (defaults to the last name on the account)*
CompanyName	*Text*	*100*	*Company name on the pickup (defaults to the company name on the account)*
SuiteOrApt	*Text*	*25*	*Suite or apartment number on the pickup (defaults to the one on file)*

Element	Type	Size or Values	Description
Address	*Text*	*100*	*Address of the pickup (defaults to the one on file)*
City	*Text*	*100*	*City of the pickup (defaults to the one on file)*
State	*Text*	*2*	*State abbreviation of the pickup (defaults to the one on file)*
ZIP5	*Numeric*	*5*	*Five digit zip code of the pickup (defaults to the one on file)*
ZIP4	*Numeric*	*4*	*+4 zip code for the pickup (defaults to the one on file)*
Phone	*Numeric*	*10*	*Phone number of the pickup (defaults to the one on file)*
Extension	*Numeric*	*5*	*Phone number extension for the pickup (defaults to the one on file)*
PackageLocation	Char	2	sd - Side Door kd - Knock on Door/Ring Bell mr - Mail Room of - Office rc - Reception im - In/At Mailbox ot - Other fd - Front Door bd - Back Door
SpecialInstructions	*Text*	*100*	*Required if "ot" chosen for PackageLocation. Special Instructions for pickup.*
PickupList	*Node*		*Shipments to be picked up.* *They can be referred to by PIC, Customs ID, or Piece ID. You only need one reference per shipment.*
PICNumber	*Numeric*	*30*	*Package PIC Number (Tracking Number)*

Element	Type	Size or Values	Description
CustomsID	*Text*	*25*	*Package Customs ID (GPM and GAPP tracked shipments only)*
PieceID	*Numeric*	*10*	*Package Piece ID*

The following listing shows an example of the CarrierPickupRequest XML for the Carrier Pickup Request API.

Listing 14a-1: CarrierPickupRequest XML

```
<?xml version="1.0" encoding="utf-8"?>
<CarrierPickupRequest>
        <AccountID></AccountID>
        <WebPassword></WebPassword>
        <Test></Test>
        <PackageLocation></PackageLocation>
        <PickupList>
                <PICNumber></PICNumber>
                <PieceID></PieceID>
                <CustomsID></CustomsID>
        </PickupList>
</CarrierPickupRequest>
```

CarrierPickupResponse XML Syntax

The following table describes the XML tags for CarrierPickupResponse.

Table 14a-2: CarrierPickupResponse XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
CarrierPickupRequestResponse	**Root**	**-**	**The root node for the XML response.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	*-*	*Error Message*
Test	*Text*	*Y or N*	*Is this a test transaction?*
Response	**Container Node**		**Container Node for the Response**
Error	*Container Node*		*Container Node For the Error (will only show IF there is an error)*
Number	*Numeric*		*USPS Error Code*
Description	*Text*	*250*	*USPS Error Code Description*
DayOfWeek	Text	Pickup Day of week (i.e. “Monday”)	Pickup day of week
Date	Date	Xx/xx/xxxx	Date of pickup
CarrierRoute	Char	1	Pickup Route type (i.e. “C” for City)

Element		Type	Size or Values	Description
	ConfirmationNumber	Text	100	Confirmation number of the pickup (Needed to change or cancel pickup)

The following listing shows an example of the CarrierPickupResponse XML.

Listing 14a-2: CarrierPickupResponse XML

```
<?xml version="1.0" encoding="UTF-8"?>
<CarrierPickupResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <Response>
                <DayOfWeek></DayOfWeek>
                <Date></Date>
                <CarrierRoute></CarrierRoute>
                <ConfirmationNumber></ConfirmationNumber>
        </Response>
</CarrierPickupResponse>
```

Using form submission: To use this API with a form submission you must declare both the method and the XML's parameter name in the submission.

Example:

```
<form name="CarrierPickupRequest"
      method="POST"
      action="URL TO API&method=CarrierPickupRequest">
      <input type="hidden" name="XMLInput" value="XML (Formatted as above)">
</form>
```

OR (through cURL in PHP):

```
$xml = XML (Formatted as above);
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, 'URL TO API');
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS,
'&method=CarrierPickupRequest&XMLInput=' . $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');

$soapclient = new soapclient('URL TO API');
```

$xml = XML (Formatted as above);

```
$params = array(
        'XMLInput' => $xml
                );

$soapclient->call('CarrierPickupRequest',$params);

echo $soapclient->response;
```

CarrierPickupRequest Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
XMLInput Error	XML is not formatted correctly.
You need at least 1 Express or Priority package	You need at least one Priority or Express mail shipment.
Invalid Package Location	You must use: sd - Side Door kd - Knock on Door/Ring Bell mr - Mail Room of - Office rc - Reception im - In/At Mailbox ot - Other fd - Front Door bd - Back Door
USPS Error Codes	**No complete list is available.**

14B. CANCELING A CARRIER PICKUP REQUEST API

CarrierPickupCancel

Table 14b-1: CarrierPickupCancel XML Elements

Element	Type	Size or Values	Description
CarrierPickupCancel	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
Test	*Text*	*Y or N*	*Is this a test transaction?*
UseAddressOnFile	*Text*	*Y or N*	*Use your physical address we have on file (Default Y)*
CompanyName	*Text*	*100*	*Company name on the pickup (defaults to the company name on the account)*
SuiteOrApt	*Text*	*25*	*Suite or apartment number on the pickup (defaults to the one on file)*
Address	*Text*	*100*	*Address of the pickup (defaults to the one on file)*
City	*Text*	*100*	*City of the pickup (defaults to the one on file)*

Element	Type	Size or Values	Description
State	*Text*	*2*	*State abbreviation of the pickup (defaults to the one on file)*
ZIP5	*Numeric*	*5*	*Five digit zip code of the pickup (defaults to the one on file)*
ZIP4	*Numeric*	*4*	*+4 zip code for the pickup (defaults to the one on file)*
ConfirmationNumber	Text	100	Confirmation number for the pickup

The following listing shows a syntax example of the CarrierPickupCancel XML for the Carrier Pickup Cancel API.

Listing 14b-1: CarrierPickupCancel XML

```
<?xml version="1.0" encoding="utf-8"?>
<CarrierPickupCancel>
        <AccountID></AccountID>
        <WebPassword></WebPassword>
        <Test></Test>
        <ConfirmationNumber></ConfirmationNumber>
</CarrierPickupCancel>
```

CarrierPickupCancelResponse XML Syntax

The following table describes the XML tags for CarrierPickupCancelResponse.

Table 14b-2: CarrierPickupCancelResponse XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
CarrierPickupCancelResponse	**Root**	**-**	**The root node for the XML response.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	*-*	*Error Message*
Test	*Text*	*Y or N*	*Is this a test transaction?*
Response	**Container Node**		**Container Node for the Response**
Error	*Container Node*		*Container Node For the Error (will only show IF there is an error)*
Number	*Numeric*		*USPS Error Code*

Element		Type	Size or Values	Description
	Description	*Text*	*250*	*USPS Error Code Description*
ConfirmationNumber		Text	100	Confirmation Number of the Pickup (Needed to change or cancel pickup)
Status		Text	250	Status of the pickup (i.e. “Cancelled”)

The following listing shows an example of the CarrierPickupCancelResponse XML.

Listing 14b-2: CarrierPickupCancelResponse XML

```
<?xml version="1.0" encoding="UTF-8"?>
<CarrierPickupResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <Response>
                <DayOfWeek></DayOfWeek>
                <Date></Date>
                <CarrierRoute></CarrierRoute>
                <ConfirmationNumber></ConfirmationNumber>
        </Response>
</CarrierPickupResponse>
```

Using form submission: To use this API with a form submission you must declare both the method and the XML’s parameter name in the submission.

Example:

```
<form name=”CarrierPickupCancel”
      method=”POST”
      action=”URL TO API&method=CarrierPickupCancel”>
      <input type=”hidden” name=”XMLInput” value=”XML (Formatted as above)”>
</form>
```

OR (through cURL in PHP):

```
$xml = XML (Formatted as above);
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, ‘URL TO API’);
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS,
‘&method=CarrierPickupCancel&XMLInput=' . $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');
$soapclient = new soapclient('URL TO API');
```
$xml = XML (Formatted as above);
```
$params = array(
        'XMLInput' => $xml
                );
$soapclient->call('CarrierPickupCancel',$params);
echo $soapclient->response;
```

CarrierPickupCancel Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
XMLInput Error	XML is not formatted correctly.
USPS Error Codes	**No complete list available.**

14C. CHANGING A CARRIER PICKUP REQUEST API

CarrierPickupChange

Table 14c-1: CarrierPickupChange XML Elements

Element	Type	Size or Values	Description
CarrierPickupChange	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
Test	*Text*	*Y or N*	*Is this a test transaction?*
UseAddressOnFile	*Text*	*Y or N*	*Use your physical address we have on file (Default Y)*
FirstName	*Text*	*100*	*First name on the pickup (defaults to the first name on the account)*
LastName	*Text*	*100*	*Last name on the pickup (defaults to the last name on the account)*
CompanyName	*Text*	*100*	*Company name on the pickup (defaults to the company name on the account)*
SuiteOrApt	*Text*	*25*	*Suite or apartment number on the pickup (defaults to the one on file)*

Element	Type	Size or Values	Description
Address	*Text*	*100*	*Address of the pickup (defaults to the one on file)*
City	*Text*	*100*	*City of the pickup (defaults to the one on file)*
State	*Text*	*2*	*State abbreviation of the pickup (defaults to the one on file)*
ZIP5	*Numeric*	*5*	*Five digit zip code of the pickup (defaults to the one on file)*
ZIP4	*Numeric*	*4*	*+4 zip code for the pickup (defaults to the one on file)*
Phone	*Numeric*	*10*	*Phone number of the pickup (defaults to the one on file)*
Extension	*Numeric*	*5*	*Phone number extension for the pickup (defaults to the one on file)*
PackageLocation	Char	2	sd - Side Door kd - Knock on Door/Ring Bell mr - Mail Room of - Office rc - Reception im - In/At Mailbox ot - Other fd - Front Door bd - Back Door
SpecialInstructions	*Text*	*100*	*Required if "ot" is chosen for PackageLocation. Special Instructions for pickup.*
ConfirmationNumber	Text	100	The confirmation number of the pickup you are changing
PickupList	**Node**	**-**	**Container node for the Carrier Pickup List**

Element	Type	Size or Values	Description
PICNumber	*Numeric*	*30*	*Package PIC Number (Tracking Number)*
CustomsID	*Text*	*25*	*Package Customs ID (GPM and GAPP tracked shipments only)*
PieceID	*Numeric*	*10*	*Package Piece ID*

The following listing shows an example of the CarrierPickupChange XML for the Carrier Pickup Change API.

Listing 14c-1: CarrierPickupChange XML

```
<?xml version="1.0" encoding="utf-8"?>
<CarrierPickupChange>
        <AccountID></AccountID>
        <WebPassword></WebPassword>
        <Test></Test>
        <PackageLocation></PackageLocation>
        <ConfirmationNumber></ConfirmationNumber>
        <PickupList>
                <PICNumber></PICNumber>
                <PieceID></PieceID>
                <CustomsID></CustomsID>
        </PickupList>
</CarrierPickupChange>
```

CarrierPickupChangeResponse XML Syntax

The following table describes the XML tags for CarrierPickupChangeResponse.

Table 14c-2: CarrierPickupChangeResponse XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
CarrierPickupChangeResponse	**Root**	**-**	**The root node for the XML response.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	*-*	*Error Message*
Test	*Text*	*Y or N*	*Is this a test transaction?*
Response	**Container Node**		**Container Node for the Response**
Error	*Container Node*		*Container Node For the Error (will only show IF there is an error)*
Number	*Numeric*		*USPS Error Code*
Description	*Text*	*250*	*USPS Error Code Description*

Element		Type	Size or Values	Description
	DayOfWeek	Text	Pickup Day of week (i.e. “Monday”)	Pickup day of week
	Date	Date	Xx/xx/xxxx	Date of pickup
	CarrierRoute	Char	1	Pickup Route type (i.e. “C” for City)
	ConfirmationNumber	Text	100	Confirmation number of the pickup (Needed to change or cancel pickup)

The following listing shows an example of the CarrierPickupChangeResponse XML.

Listing 14c-2: CarrierPickupChangeResponse XML

```
<?xml version="1.0" encoding="UTF-8"?>
<CarrierPickupChangeResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <Response>
                <DayOfWeek></DayOfWeek>
                <Date></Date>
                <CarrierRoute></CarrierRoute>
                <ConfirmationNumber></ConfirmationNumber>
        </Response>
</CarrierPickupChangeResponse>
```

Using form submission: To use this API with a form submission you must declare both the method and the XML’s parameter name in the submission.

Example:

```
<form name=”CarrierPickupChange”
      method=”POST”
      action=”URL TO API&method=CarrierPickupChange”>
      <input type=”hidden” name=”XMLInput” value=”XML (Formatted as above)”>
</form>
```

OR (through cURL in PHP):

$xml = *XML (Formatted as above)*;

```
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, ‘URL TO API’);
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS,
‘&method=CarrierPickupChange&XMLInput=' . $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
```

```
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');
$soapclient = new soapclient('URL TO API');
```

$xml = *XML (Formatted as above)*;

```
$params = array(
        'XMLInput' => $xml
                );
$soapclient->call('CarrierPickupChange',$params);
echo $soapclient->response;
```

CarrierPickupChange Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
XMLInput Error	XML is not formatted correctly.
You need at least 1 Express or Priority package	You need at least one Priority or Express mail shipment.
Invalid Package Location	You must use: sd - Side Door kd - Knock on Door/Ring Bell mr - Mail Room of - Office rc - Reception im - In/At Mailbox ot - Other fd - Front Door bd - Back Door
USPS Error Codes	**No complete list available.**

14D. SCHEDULING A MULTIPLE LOCATION CARRIER PICKUP REQUEST API

MultiLocationCarrierPickup

Table 14d-1: MultiLocationCarrierPickup XML Elements

Element	Type	Size or Values	Description
MultiLocationCarrierPickup	**Root**	**-**	**The root node for the XML request.**
AccountID	Numeric	6	Account ID for the Endicia postage account.
WebPassword	Text	17	Customer's Web Password. **Important!** You **must** use either the WebPassword or the PassPhrase element.The WebPassword element is **optional** if you use the PassPhrase element.
PassPhrase	Text	64	Customer's Pass Phrase **Important!** You **must** use either the PassPhrase or the WebPassword element. The PassPhrase element is **optional** if you use the WebPassword element.
Test	*Text*	*Y or N*	*Is this a test transaction?*

The following listing shows a syntax example of the CarrierPickupRequest XML for the Carrier Pickup Request API.

Listing 14d-1: MultiLocationCarrierPickup XML

```
<?xml version="1.0" encoding="utf-8"?>
<MultiLocationCarrierPickup>
        <AccountID></AccountID>
        <WebPassword></WebPassword>
        <Test></Test>
</MultiLocationCarrierPickup>
```

MultiLocationCarrierPickupResponse XML Syntax

The following table describes the XML tags for CarrierPickupResponse.

Table 14d-2: MultiLocationCarrierPickupResponse XML Elements (XMLInput Parameter)

Element	Type	Size or Values	Description
MultiLocationCarrierPickupResponse	**Root**	**-**	**The root node for the XML response.**

Element	Type	Size or Values	Description
AccountID	Numeric	6	Account ID for the Endicia postage account.
ErrorMsg	*Text*	-	*Error Message*
Test	*Text*	*Y or N*	*Is this a test transaction?*
ScheduledPickups	**Container Node**		**Container Node for the Scheduled Pickups**
GroupID	Text		Cost Center of the Pickup / Container Node of Pickup
CompanyName	Text		Company of pickup
Address	Text		Address of pickup
Name	Text		Name of person at pickup
City	Text		City of pickup
State	Char	2	State code of pickup
ZipCode	Numeric		ZIP Code of pickup
PhoneNumber	Text		Phone Number of pickup
EmailAddress	Text		E-mail address of pickup
ConfirmationNumber	Text	100	Confirmation number of pickup
Message	Text		Date of pickup
NotScheduled	**Container Node**		**Container node for the pickups not scheduled**
GroupID	Text		Cost center of the missed pickup / container node of missed pickup
CompanyName	Text		Company of missed pickup
Address	Text		Address of missed pickup
Name	Text		Name of person at missed pickup
City	Text		City of missed pickup
State	Char	2	State code of missed pickup
ZipCode	Numeric		ZIP Code of missed pickup
PhoneNumber	Text		Phone Number of missed pickup
EmailAddress	Text		E-mail address of missed pickup
ConfirmationNumber	Text	100	Confirmation number of missed pickup

Element		Type	Size or Values	Description
	Message	Text		Reason why it was not scheduled

The following listing shows an example of the MultiLocationCarrierPickupResponse XML.

Listing 14d-2: MultiLocationCarrierPickupResponse XML

```
<?xml version="1.0" encoding="UTF-8"?>
<MultiLocationCarrierPickupResponse>
        <AccountID></AccountID>
        <ErrorMsg></ErrorMsg>
        <Test></Test>
        <ScheduledPickups>
                <GroupID>
                        <Name></Name>
                        <CompanyName></CompanyName>
                        <Address></Address>
                        <City></City>
                        <State></State>
                        <ZipCode></ZipCode>
                        <PhoneNumber></PhoneNumber>
                        <EmailAddress></EmailAddress>
                        <ConfirmationNumber></ConfirmationNumber>
                        <Message></Message>
                </GroupID>
        </ScheduledPickups>
        <NotScheduled>
                <GroupID>
                        <Name></Name>
                        <CompanyName></CompanyName>
                        <Address></Address>
                        <City></City>
                        <State></State>
                        <ZipCode></ZipCode>
                        <PhoneNumber></PhoneNumber>
                        <EmailAddress></EmailAddress>
                        <ConfirmationNumber></ConfirmationNumber>
                        <Message></Message>
                </GroupID>
        </NotScheduled>
</MultiLocationCarrierPickupResponse>
```

Using form submission: To use this API with a form submission you must declare both the method and the XML's parameter name in the submission.

Example:

```
<form name="MultiLocationCarrierPickupRequest"
      method="POST"
      action="URL TO API&method=MultiLocationCarrierPickup">
      <input type="hidden" name="XMLInput" value="XML (Formatted as above)">
</form>
```

OR (through cURL in PHP):

$xml = *XML (Formatted as above)*;

```
$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL, 'URL TO API');
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS,
'&method=MultiLocationCarrierPickup&XMLInput=' . $xml);
$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");
curl_close ($curl_handle);
echo $curl_result;
```

Example using SOAP submission (with PHP)

```
require_once('nusoap.php');

$soapclient = new soapclient('URL TO API');
```

$xml = *XML (Formatted as above)*;

```
$params = array(
        'XMLInput' => $xml
                );

$soapclient->call('MultiLocationCarrierPickup',$params);


echo $soapclient->response;
```

CarrierPickupRequest Error Codes

Error Code String	Error Message
Login Failed	Given in the root node, occurs if the account number or password is incorrect.
XMLInput Error	XML is not formatted correctly.
Too Few Priority or Express	You need at least one Priority or Express mail shipment.
USPS Error Codes (Message Node)	**No complete list available.**

15. ADDRESS VALIDATION–DIAL-A-ZIP

The Endicia Label Server Web Service requires a valid destination address in the LabelRequest XML to process a shipping label. To ensure that your address is in the format that the USPS requires, refer to USPS Publication 28, Postal Addressing Standards, which you can download from:

http://pe.usps.com/cpim/ftp/pubs/Pub28/pub28.pdf

If an address has been rejected, see the error message in the LabelRequestResponse XML to correct the address being submitted.

Note: The Endicia Label Server calls the Dial-A-ZIP servers to validate addresses. **If you want to "advance validate" or process addresses that are *not* bound for the Endicia Label Server, contact us for a Dial-A-ZIP address validation API.**

The Endicia Label Server Web Service returns a label if the destination address:

- is an exact match.
- has a valid city, state, and ZIP Code, but the street address is not a match.
- is a default match in the Dial-A-ZIP address database.

The Endicia Label Server Web Service will ***not*** return a label if the destination address:

- is invalid.
- has an invalid city, state, or ZIP Code.
- is not found.
- has multiple matches, or is too ambiguous to resolve to a default address.

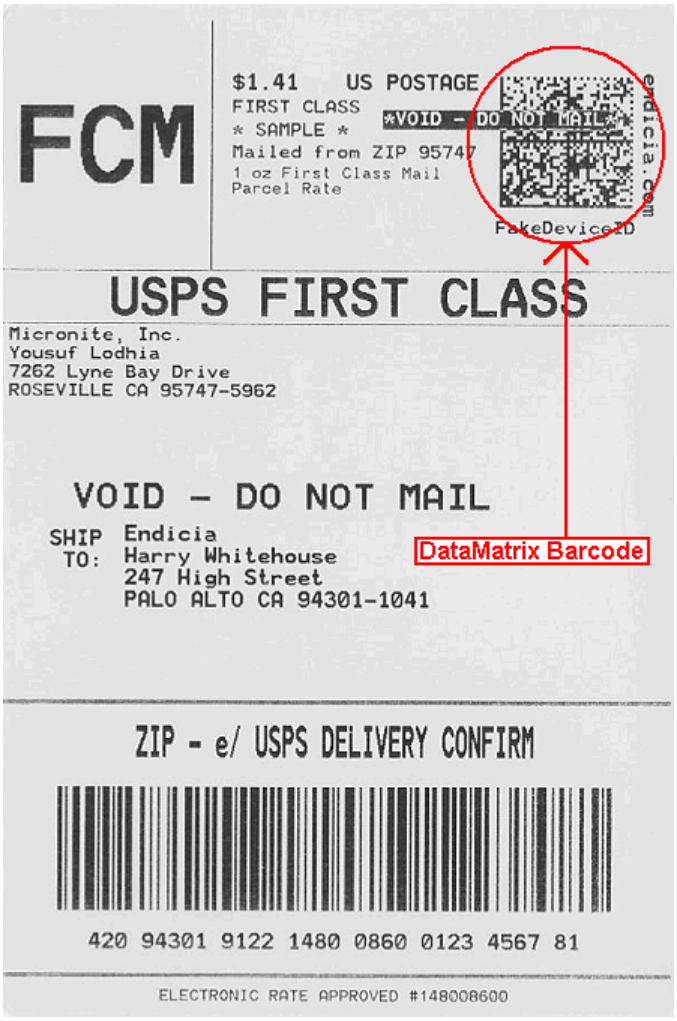
16. PRINTING USING EPL2 AND ZPLII

For certain label layouts, the Label Server Web Service can generate label images in Zebra's native printer languages, EPL2 and ZPLII. The label image can contain data in both text and binary. To ensure that the USPS can successfully scan the barcodes in the label, care must be taken to preserve the raw printer data when sending the label image to the Zebra printer.

Some guidelines to follow when printing EPL2 and ZPLII labels to Zebra printers include:

- Verify that your Zebra printer has the appropriate DataMatrix firmware installed

  The Label Server uses the DataMatrix symbology. If the Zebra printer does not contain the appropriate firmware, the square DataMatrix postage barcode would not print on the label and the USPS will reject the mailpiece. The following is an EPL2 label with the DataMatrix barcode in the upper-right corner:



  To upgrade the printer's firmware to support DataMatrix barcodes, visit the Zebra support Web site:

  http://www.zebra.com/id/zebra/na/en/index/drivers_downloads/firmware/firmware_for_printers.html

- Treat all data contained in the EPL2 or ZPLII label image as binary bytes

  The label image contains text and binary data values. All data must be treated as binary. Therefore, whether sending the label image data directly to the printer or saving it to intermediate storage for later retrieval, there should be no conversion performed to ANSI, ASCII or any other format. If that happens, the crucial barcode data bits will be changed causing the USPS scanners to read erroneous barcode data.

  The following example in Visual C# shows how to send raw data to a printer:

```
public static bool SendBytesToPrinter(string szPrinterName, byte[] bytes)
{
  Int32     nLength;
  IntPtr    pUnmanagedBytes;
  bool      bSuccess;

  pUnmanagedBytes = new IntPtr(0);
  nLength = Convert.ToInt32(bytes.Length);

  pUnmanagedBytes = Marshal.AllocCoTaskMem(nLength);

  Marshal.Copy(bytes, 0, pUnmanagedBytes, nLength);

  bSuccess = SendBytesToPrinter(szPrinterName, pUnmanagedBytes, nLength);

  Marshal.FreeCoTaskMem(pUnmanagedBytes);

  return(bSuccess);
}

public static bool SendBytesToPrinter(string szPrinterName,
                                      IntPtr pUnmanagedBytes, Int32 nLength)
{
  DOCINFOA  di;
  Int32     dwError, dwWritten;
  IntPtr    hPrinter;
  bool      bSuccess;

  bSuccess = false;

  dwError = 0;
  dwWritten = 0;

  di = new DOCINFOA();
  di.pDocName = "ELS Label";
  di.pDataType = "RAW";

  hPrinter = new IntPtr(0);
  if (OpenPrinter(szPrinterName.Normalize(), out hPrinter, IntPtr.Zero) == true)
  {
     if (StartDocPrinter(hPrinter, 1, di) == true)
     {
        if (StartPagePrinter(hPrinter))
        {
            bSuccess = WritePrinter(hPrinter, pUnmanagedBytes, nLength,
                                    out dwWritten);
            EndPagePrinter(hPrinter);
        }
        EndDocPrinter(hPrinter);
     }
     ClosePrinter(hPrinter);
  }
```

```
    if (bSuccess == false)
    {
       dwError = Marshal.GetLastWin32Error();
    }

    return(bSuccess);
}
```

- Correctly decode the sequence of data bytes in the label image

  The Zebra EPL2 and ZPLII languages have special sequences of bytes to represent special characters, such as a quote, back slash, and so on. These special sequences must be recognized and decoded properly to obtain a scannable label. Refer to the Zebra programming guides for further information.

17. MISCELLANEOUS NOTES AND FEATURES

Topics in this section include:

- Bulk Acceptance Scans (SCAN)
- Automated Carrier Pickup
- Endicia Printable Postage XML
- APO/FPO Addresses
- Proxy Server
- Accessing the Endicia Label Server Web Service through SSL/HTTPS
- Eligible Matter for Delivery or Signature Confirmation
- Express Mail Sunday and Holiday Service
- Rubber Stamps

Bulk Acceptance Scans (SCAN)

SCAN feature is available with each Endicia Label Server account through Endicia.com. This feature links all daily shipments with either Delivery or Signature Confirmation to a single bar-coded form that the USPS scans when packages are picked up. When this scan data is uploaded into the postal system that night, all your packages will be marked accepted on the USPS tracking system.
When your customer verifies the status of their package instead of reading that the USPS was electronically notified of the item, your customer will see "The USPS accepted this item at Date and Time".

http://www.endicia.com/Features/BulkAcceptanceScans/

Automated Carrier Pickup

Endicia has automated USPS “Carrier Pickup”. This feature can be used to schedule next day pickup by the Postal Service for packages created by the Endicia Label Server (Express Mail, Priority Mail and International shipments). More information is available upon request.

Endicia Printable Postage XML

The *Endicia Printable Postage* solution is available with your Endicia Label Server accounts. Endicia Printable Postage can print stamps without entering any address information. While Endicia Printable Postage is a client based program, Endicia has an XML interface for Endicia Printable Postage rolls, which can be printed through DYMO printers. This interface can be used in conjunction with Endicia Label Server to address various light weight mail needs. More information is available upon request.
http://www.endicia.com/Features/InstaPostage/

APO/FPO Addresses

Currently, the USPS does not allow Signature Confirmation to be used for APO/FPO addresses. Delivery Confirmation is to restricted APO/FPO destinations.

Proxy Server

If the Endicia Label Server Web Service is being called from behind a proxy server that enforces limits on SOAP packet size, you may experience issues with sending and receiving large SOAP packets. For example, a call to the Endicia Label Server Web Service to render a label with a high resolution results in a large SOAP packet being sent. You may need to bypass the proxy, set the SOAP packet size limit very high, or limit the amount of data you are requesting.

Accessing the Endicia Label Server Web Service through SSL/HTTPS

Because the Endicia Label Server Web Service resides on an SSL-enabled server, it will present a digital certificate to the client for authentication. The client must *explicitly* trust the certificate in order for the connection to be set up. If on the other hand, the client chooses not to trust the certificate, the connection will fail.

For further discussion and resolution of this issue, including a suggested workaround in.Net, see Article #823177 on Microsoft's support Web site at:

http://support.microsoft.com/default.aspx?scid=kb;en-us;823177

Eligible Matter for Delivery or Signature Confirmation

Delivery or Signature Confirmation is available for First-Class Mail parcels, Package Services parcels and for all Priority Mail pieces. For further details, consult the USPS Domestic Mail Manual, Section 503 Extra Services (http://pe.usps.gov/text/dmm300/503.htm).

Express Mail Sunday and Holiday Service

On March 3, USPS launched a new Express Mail Sunday and Holiday Delivery service. For an additional fee of $12.50, your Express Mail mailpiece will be delivered on Sundays and Federal holidays.

Express Mail Sunday and Holiday delivery service is not available to all destinations.

When shipping a mailpiece with the MailClass of ExpressMail and SundayHolidayDelivery set to true, the Endicia Label Server performs the following steps:

1. Verify if the next day or second day after the ShipDate is on a Sunday or a Holiday. If it is not, the Endicia Label Server returns error code 11111.

2. Retrieve the Express Mail Service Commitments from the USPS servers using the origin and destination ZIP Codes and ShipDate.
   - If no Service Commitments are received for Next Day 12 PM, Next Day 3PM or 2-Day delivery service, the Endicia Label Server returns error code 11112.
   - If Service Commitments are received for two delivery days only, then the Endicia Label Server returns error code 11113.
   - In case of an error communicating with the USPS servers, the Endicia Label Server can return either error code 11102, ExpressMailAPIError, or error code 11110.

3. Verify the day when the Service Commitment occurs. If it does not fall on a Sunday or a Holiday, the Endicia Label Server returns error code 11113.

4. Finally, cycle through all locations for the Service Commitments to determine if at least one location accepts Express Mail shipments past the supplied ShipTime. If all cut-off times have expired, the Endicia Label Server returns error code 11114. ShipTime is optional; if it is not supplied, it defaults to 12:01 AM, essentially accepting any location in this step.

5. For a label or price request, the Endicia Label Server returns the appropriate response with the $12.50 Sunday-Holiday Delivery premium included in the postage. For a label request, if the

destination ZIP Code does not deliver Express Mail on Sundays or Holidays, the Endicia Label Server returns error code 11115.

Rubber Stamps

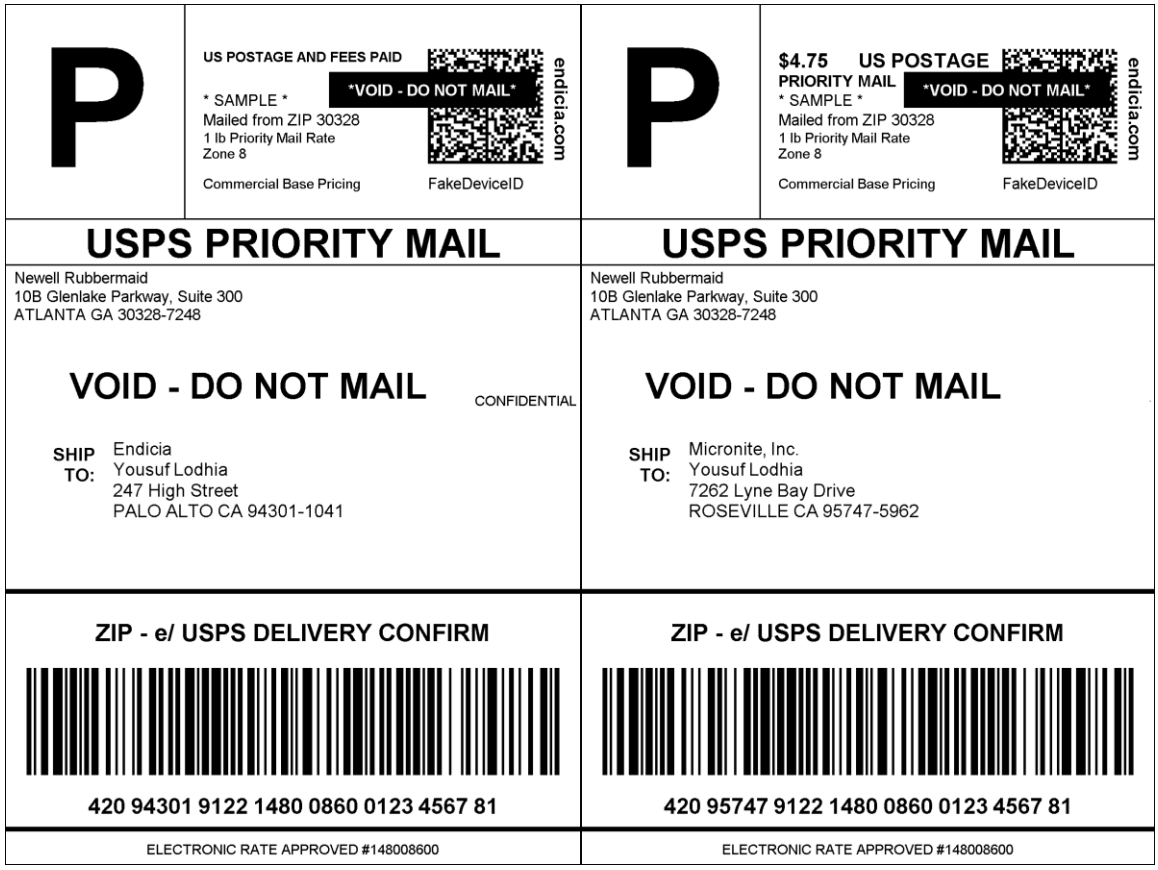
A label can contain one or more user-supplied text that are called *Rubber Stamps*. The Endicia Label Server Web Service supports up to three Rubber Stamp fields within a label. Currently, the Rubber Stamp fields are available for the default label type in the following label sizes:

- 4×6
- 4×5
- 4×4.5
- DocTab

Express Mail, international, and integrated labels do not contain Rubber Stamp fields.

Although each Rubber Stamp field can have as many as 50 characters, the actual number of characters that can be printed on the label depends on the content of the Rubber Stamp text. This is true for all labels which are not in EPL2 format because all text is drawn using proportional fonts. Labels in EPL2 format, however, use monospaced fonts. Therefore, the number of characters in the Rubber Stamp fields will be much less than the maximum allowed. For example, a 4×6 default label in EPL2 format can accommodate 26 or fewer characters in each Rubber Stamp field.

The label on the left is a Stealth postage label, which does not show the postage price. The label on the right shows the postage price.



18. FREQUENTLY ASKED QUESTIONS

Q. What does a customer need to do to move to the production environment?
A. A customer needs to send:
- A sample label that the customer has produced from the Test Server. Send the sample label by e-mail to LabelServer@endicia.com.
- A signed agreement, including Exhibit B. **Note:** The agreement was sent to the customer by e-mail, along with the technical specifications.

Q. How do I contact technical support?
A. Support is available by e-mail message through LabelServer@endicia.com. If a phone call is required we can schedule one at that time. **Note**: Include your XML request in any technical support requests when it is available.

Q. How do I purchase postage in production and what are my postage payment options?
A. You must purchase postage through the API by using the RecreditRequest method. To review your postage payment options, go to the following URL: http://www.endicia.com/USPSPaymentOptions.cfm

Q. How do I reset my Pass Phrase in production?
A. You must reset the Pass Phrase through the API by using the ChangePassPhraseRequest method. **Note:** Make sure your developer adds this feature because after you sign up for a new Endicia Label Server account, you will be required to reset the Pass Phrase for security purposes.

Q. If my account gets suspended due to bad login attempts, how can I reactivate my account?
A. To reactivate your account, send an e-mail message to support at LabelServer@endicia.com, and then reset the Pass Phrase through the ChangePassPhraseRequest API.

Q. Do you offer a reprint option?
A. Reprinting is not available from the Endicia Label Server. You can either store the label image on your side or request a refund on the cancelled label and reprint.

Q. What are the requirements for submitting a refund for misprints or unused mail pieces?
A. All labels using Delivery Confirmation, Signature Confirmation, or the Express Mail tracking number must be submitted electronically. No physical submissions are allowed. To qualify, the item cannot be scanned by the USPS and must be requested within 10 days of printing.

Q. Having problems with printing the barcode on the label using Zebra printer?
A. If you are using the Zebra LP-2844 printer, you must upgrade the printer's firmware to support DataMatrix barcodes.

Q. Do you offer customizable notification e-mail options such as DAZzle?
A. E-mail notifications are only available for Express Mail Services. However, because we return the tracking number, you can easily send out the same exact e-mail notification based on your own business rules.

Q. What are the options for shipping internationally?
A. First-Class Mail International, Priority Mail International, and Express Mail International are available for international shipping.

Q. What can you tell me about shipping to APO/FPO addresses that result in "Address not found"?

A. APO/FPO addresses are generally problematic. There is no official format that we are aware of. Missing or extra punctuation may cause addresses to be rejected by the address validation engine. For more information, see URL: http://www.penpalsforsoldiers.org/address.html

Q. Do APO/FPO shipments require customs forms?

A. According to the USPS, all mail going to an APO/FPO containing merchandise should bear Customs Forms 2976-A. For more information, see: http://webapps.usps.com/customsforms/helppickaform.htm

Q. When are dimensions required when submitting label requests?

A. Dimensions are required for Priority Mail items destined for Zones 5-8.

Q. Can I add my company logo to the shipping label?

A. We do not offer it as part of the interface, partly because the overhead of sending the additional image back and forth may impact performance. However, we return your image of the label, on which you can superimpose your own logo.

Q. Do you have a suggested site where we can send customers to view the status of their tracking number?

A. Here are two links:

- http://www.endicia.com/Status/?PIC=TRACKINGNUMBERHERE
- http://trkcnfrm1.smi.usps.com/PTSInternetWeb/InterLabelInquiry.do?origTrackNum=TRACKINGNUMBERHERE

Note: The second URL has much more current information.

Q. How do I convert a string containing the label image in Base64 format returned in the LabelRequestResponse XML into a bitmap object in .Net?

A. The following is a function in C# that converts a string containing the label image in Base64 format returned in the label request response into a bitmap object you can draw on the screen or print:

```
private Bitmap _GetBitmap(string base64LabelImage)
{
    byte[]       bytes = null;
    Bitmap       labelBitmap;
    MemoryStream memoryStream;
    bytes = Convert.FromBase64String(base64LabelImage);
    memoryStream = new MemoryStream(bytes);
    labelBitmap = (Bitmap) Image.FromStream(memoryStream);
    memoryStream.Close();
    memoryStream = null;
    return(labelBitmap);
}
```

Q. Is there a way to ensure that I have a postage account balance so that my account does not ever run out of funds?

A. The best way to ensure that you do not run out of funds is to write a statement to view the postage account balance that is included with every LabelRequestResponse, and if that postage account balance value is less than a specified dollar amount, subtract that postage account balance value from your postage account limit, and send a RecreditRequest with the amount being the result.

For example, if the postage account value is less than $100, subtract that value from $500 and submit the RecreditRequest for the difference.

19. MAILPIECE SHAPE CHART

The LabelRequest XML contains a MailpieceShape element. The following table shows valid combinations of mailpiece shapes and mail classes, grouped by label type, for use with the MailpieceShape element.

LabelType of Default									
	Mailpiece Shape								
	Card	Letter	Flat	Parcel	Flat Rate Box Small , Medium and Large	Flat Rate Envelope, Padded Envelope	Large Parcel[1]	Irregular Parcel	Oversized Parcel[2]
Express		✓	✓	✓		✓			
First				✓					
Library Mail				✓					
Media Mail				✓					
Parcel Post				✓			✓		✓
Parcel Select				✓			✓		✓
Priority			✓	✓	✓	✓	✓	✓	
Standard Mail				✓				✓	

1. Large Parcel is a parcel that weighs less than 20 pounds but measures more than 84 inches in combined length and girth (but not more than 108 inches).
2. Oversized Parcel is a parcel that weighs less than 20 pounds but measures more than 108 inches in combined length and girth (but not more than 130 inches).
3. The Flat Rate Padded Envelope is only available to qualifying Priority Mail Commercial Plus customers.

LabelType of DestinationConfirm									
	Mailpiece Shape								
	Card	Letter	Flat	Parcel	Flat Rate Box Small , Medium and Large	Flat Rate Envelope, Padded Envelope	Large Parcel	Irregular Parcel	Oversized Parcel
First	✓	✓	✓						

LabelType of CertifiedMail									
	Mailpiece Shape								
	Card	Letter	Flat	Parcel	Flat Rate Box Small , Medium and Large	Flat Rate Envelope, Padded Envelope	Large Parcel	Irregular Parcel	Oversized Parcel
First	✓	✓	✓	✓					
Priority	✓	✓	✓	✓					

LabelType of International									
	Mailpiece Shape								
	Card	Letter	Flat	Parcel	Flat Rate Box Small , Medium and Large	Flat Rate Envelope	Large Parcel[1]	Irregular Parcel	Oversized Parcel[2]
Express Mail International			✓	✓		✓			
First-Class Mail International		✓	✓	✓					
Priority Mail International			✓	✓	✓	✓			

If MailpieceShape is not supplied, it defaults to Parcel with the following exceptions:

- First-Class: DestinationConfirm and CertifiedMail labels default to Letter if WeightOz is 3.5 oz or less; otherwise, it defaults to Flat.
- Priority Mail: CertifiedMail labels defaults to Flat.

Appendix A. Change History

The following table shows the changes to the Endicia Label Server. The most recent changes appear first.

Date	Change
2010-06-18 Ver 5.0.3826	➢ Version 5.0 ▪ Added Continuation Page to international customs forms to allow up to 30 customs items. ▪ Added support for integrated labels with customs forms for international, APO/FPO, and other domestic destinations. These labels allow up to 30 customs items. ▪ **CustomsFormImageResolution** (LabelRequest XML) Deleted value: 150 (Obsolete) ▪ **CustomsInfo** (LabelRequest XML) New: To specify the customs declaration. ▪ **ImageResolution** (LabelRequest XML) Deleted value: 150 (Obsolete) ▪ **IntegratedFormType** (LabelRequest XML) New: To specify the form type to use for an integrated label. ▪ **LabelSubtype** (LabelRequest XML) New: To specify a subtype of a label type. ▪ **LabelType** (LabelRequest XML) Added new value: Domestic ▪ **ToCountryCode** (LabelRequest XML, PostageRateRequest XML, PostageRatesRequest XML) New: To specify the country code for an international address. ▪ Updated description of the following elements in the LabelRequest XML: **CustomsFormType, CustomsFormImageFormat,** **CustomsFormImageResolution** **OriginCountry, ContentsType, ContentsExplanation**, **NonDeliveryOption** **CustomsSigner** **ImageFormat** **ReplyPostage**, **ShowReturnAddress, TrackingNumber** **ReturnAddress3, ReturnAddress4, ToAddress3, ToAddress4** **ToCountry** ▪ Added new error codes: 1002 ElementNotAllowed 1050 NoCustomsItems 1055 ExceedsValueLimitForCustomsForm 1505 MailClassNotAllowedForLabel 11303 NoLabelNumberReceived 11304 InvalidLabelNumberReceived

Date	Change
	Backward compatibility note: Any element or value marked as obsolete in this version will no longer be supported after March 31, 2011. If your code uses these elements or values, the Endicia Label Server Web Service will continue to accept them until March 31, 2011.
2010-04-07 Ver 4.9.3741	➢ Version 4.9 ➢ **CalculatePostageRates, CalculatePostageRatesXML** Added new Web methods to calculate "all" domestic or international postage rates.
2010-02-09 Ver 4.8.3691	➢ Version 4.8 ▪ Added support for the updated USPS international and customs forms 2976-A. ▪ **EelPfc** (LabelRequest XML) New: To specify the EEL/PFC value for the customs forms 2976-A.
2009-12-24 Ver 4.7.3642	➢ Version 4.7 ▪ Added support for USPS rates effective January 4, 2010. ▪ **MailpieceShape** (LabelRequest XML, PostageRateRequest XML) Added new value: FlatRatePaddedEnvelope ▪ **SortType** (LabelRequest XML, PostageRateRequest XML) Added new value: Nonpresorted. ▪ **EntryFacility** (LabelRequest XML, PostageRateRequest XML) Updated description. ▪ **MailpieceDimensions** (LabelRequest XML, PostageRateRequest XML) Updated description. ▪ **POZipCode** (LabelRequest XML) Updated description. ▪ **InsuredMail** (LabelRequest XML, PostageRateRequest XML) Increased the maximum insurable amount for Endicia Parcel Insurance to $10,000. ▪ **Country and Locality Listing** Added Kosovo. ▪ Added new error codes: 501 NoCubicPostageCalculator 502 CubicPricingNotOffered 503 MailpieceCubicVolumeMustBePositive 504 MailpieceCubicVolumeExceedsMaximum 2020 AccountNotQualifiedForMailClass 2021 AccountNotQualifiedForFeature

Date	Change
2009-11-17 Ver 4.5.3603	➢ Version 4.5 ▪ **MailClass** (LabelRequest XML, PostageRateRequest XML) Added new value: StandardMail ▪ **SortType** (LabelRequest XML, PostageRateRequest XML) Added new values: BMC, MixedBMC, SCF. ▪ **EntryFacility** (LabelRequest XML, PostageRateRequest XML) Updated description. ▪ **FromPostalCode** (PostageRateRequest XML) Updated description. ▪ **POZipCode** (LabelRequest XML) Updated description. ▪ **SundayHolidayDelivery** (LabelRequest XML, PostageRateRequest XML) New: To request Express Mail Sunday/Holiday Delivery Service. ▪ **ServiceLevel** (LabelRequest XML, PostageRateRequest XML) Deleted value: SundayHolidayDelivery ***Backward compatibility note:*** When ServiceLevel contains a value of SundayHolidayDelivery, the SundayHolidayDelivery element will be set to true.
2009-09-17 Ver 4.4.3544	➢ Version 4.4 ▪ **International Mail labels (LabelRequest XML)** The tracking number for all international mail labels will be generated by Endicia. ▪ **FromName, FromCompany, ToName, ToCompany** Updated description. ▪ **SCAN Form API** The SCAN Form API now supports international labels with Endicia-generated tracking numbers.
2009-07-17 Ver 4.3.3485	➢ Version 4.3 ▪ Added support for USPS NSA rates for qualified accounts. ▪ **FromPhone** (LabelRequest XML) Updated: Made required for international mail. ▪ **MailpieceShape** (LabelRequest XML, PostageRateRequest XML) Added new value: MediumFlatRateBox Deleted value: FlatRateBox As of May 2009, USPS has renamed the regular Priority Mail Flat Rate box to Medium Flat Rate box. ***Backward compatibility note:*** When MailpieceShape contains FlatRateBox, it will be changed to MediumFlatRateBox. ▪ **ToPhone** (LabelRequest XML) Updated: Made optional for international mail. ▪ Added new error codes:

Date	Change
	11002 PICNumberServiceError 11003 InvalidPICNumberReceived 11202 PlanetCodeNumberServiceError 11203 InvalidPlanetCodeNumberReceived 11451 FailedRetrievingAccountInformation 11452 AccountInformationServiceError 11453 InvalidValueInAccountInformation 11503 UnknownDialAZipReturnCode
2009-05-03 Ver 4.2.3404	➢ Version 4.2 ▪ Added support for USPS domestic and international rates effective May 11, 2009. ▪ **LabelSize** (LabelRequest XML) To comply with USPS requirements, added a FIM to the indicia in EnvelopeSize10 and Postcard. ▪ **Services, ElectronicReturnReceipt** (LabelRequest XML) **BpodClientDUNSNumber** (LabelRequest XML) Updated description. ▪ **Stealth** (LabelRequest XML). Stealth cannot be used with LabelSize of EnvelopeSize10 and Postcard. ▪ **EmailOptions** (LabelRequest XML) Obsolete.
2009-03-20 Ver 4.1.3363	➢ Version 4.1 ▪ **Express Mail labels (LabelRequest XML)** The tracking number for all domestic Express Mail labels will be generated by Endicia. ▪ **Express Mail shipments** (SCAN Form Web Service API) SCAN form now accepts domestic Express Mail tracking numbers. ▪ **TrackingNumber** (LabelRequestResponse XML) Updated description.
2009-02-28 Ver 4.1.3344	➢ Version 4.1 ▪ **CODAmount** (LabelRequest XML, PostageRateRequest XML) New: To specify COD amount. ▪ **EmailOptions** (LabelRequest XML) Obsolete: Beginning March 15th, 2009, Endicia Label Server will no longer support e-mail notifications for Express Mail shipments. ▪ **EntryFacility** (LabelRequest XML, PostageRateRequest XML) New: To specify the entry facility. ▪ **FromPostalCode** (PostageRateRequest XML) Updated description. ▪ **InsuredValue** (LabelRequest XML, PostageRateRequest XML)

Date	Change
	Updated description. ▪ **MailClass** (LabelRequest XML, PostageRateRequest XML) Added new value: ParcelSelect ▪ **POZipCode** (LabelRequest XML) Updated description. ▪ **Services** (LabelRequest XML, PostageRateRequest XML) Added new attributes: COD, RestrictedDelivery ▪ **SortType** (LabelRequest XML, PostageRateRequest XML) Added new values: FiveDigit, Presorted ▪ **BarcodeFormat**, **FlatRate, NoHolidayDelivery**, **Oversize, PackageSize, ReturnToSender, TrackingType** (LabelRequest XML): Obsolete. ▪ **FlatRate, PackageSize** (PostageRateRequest XML)**:** Obsolete. ▪ **MailClass** (LabelRequest XML, PostageRateRequest XML) Obsolete values: INTLAIRPARCEL, INTLGEM, INTLGPM ▪ Added new error codes: 409 InvalidMailpieceValue 436 InvalidEntryFacility 437 EntryFacilityDoesNotServiceDesination 11301 FailedRetrievingLabelNumber 11302 LabelNumberServiceError
2009-01-11 Ver 4.0.3296	➢ Version 4.0 ▪ Added support for USPS domestic and international rates effective January 18, 2009. ▪ **GetPostageLabel, GetPostageLabelXML** Added support for USPS Commercial Plus pricing. These Web methods will use the Commercial Plus rate to calculate the postage based on the qualification of the account. Note: Accounts that qualify for Commercial Plus Pricing have been identified by the USPS and require no change in the label request. All other accounts will use Commercial Base Pricing by default for domestic Express Mail and Priority Mail service. ▪ **CalculatePostageRate, CalculatePostageRateXML** Added support for USPS Commercial Plus pricing. A new Pricing element has been added to the PostageRateRequest XML that can be used to request different levels of pricing. ▪ **IncludePostage** (LabelRequest XML) Allow IncludePostage value of FALSE for all mail classes. This capability allows you to print labels without postage. ▪ **MailpieceShape** (LabelRequest XML, PostageRateRequest XML)

Date	Change
	Added new value: SmallFlatRateBox ▪ **Pricing** (PostageRateRequest XML) New: To select pricing level. ▪ **PostagePrice** (LabelRequestResponse XML, PostageRateResponse XML) Updated: **Pricing** element within the **Postage** node to reflect new USPS pricing. ▪ **CostCenter** (GetTransactionsListing XML) A new element within the TrackingList node of the GetTransactionsListing API called <CostCenter> that can have a value of "All" or any numeric value to pull transactions with the specific CostCenter value. This is an optional value and the Cost Center will not be included in the response unless it has been added to the request.
2008-10-30 Ver 3.55.3223	➢ Version 3.55 ▪ **MailClass** (LabelRequest XML, PostageRateRequest XML) Added new value: FirstClassMailInternational ▪ **IncludePostage** (LabelRequest XML) Allow IncludePostage value of FALSE for international mail classes. ▪ **InsuredMail** (LabelRequest XML, PostageRateRequest XML) For international mail classes, the Label Server will return an error if Endicia Insurance is not available for the destination country. ▪ Added new error codes: 422 OnlinePostageNotAllowed 448 InsuranceNotAvailableToCountry ▪ E-refunds are not currently supported for FCMI through Label Server. We will provide support for physical refunds in the next update to the Endicia Label Server.
2008-09-12 Ver 3.54.3177	➢ Version 3.54 ▪ **MailClass** (LabelRequest XML, PostageRateRequest XML) As of September 11, the USPS will require all Bound Printed Matter mailings to be paid by permit only. Therefore, BoundPrintedMatter will no longer be supported. ▪ **Stealth** (LabelRequest XML) Allow Stealth value of FALSE for international mail classes.
2008-07-22 Ver 3.53.3124	➢ Version 3.53 ▪ **NonDeliveryOption** (LabelRequest XML) New: To specify nondelivery instructions. ▪ **ReturnToSender** (LabelRequest XML) Obsolete.
2008-07-06 Ver 3.52.3103	➢ Version 3.52 ▪ **EmailOptions** (LabelRequest XML) New: To choose e-mail notification option. ▪ **ImageFormat, CustomsFormImageFormat** (LabelRequest XML)

Date	Change
	Added new value: PDF
2008-06-06 Ver 3.51.3077	➢ Version 3.51 ▪ **Postage** (PostageRateResponse XML) Updated description. ▪ **PostagePrice** (LabelRequestResponse XML, PostageRateResponse XML) New: To return the postage price. ▪ **ResponseOptions** (LabelRequest XML, PostageRateRequest XML) New: To choose XML elements returned in the response.
2008-05-03	➢ Version 3.5 Added support for USPS domestic and international rates effective May 12, 2008.
2008-04-30 Ver 3.4.3041	➢ Version 3.4 ▪ **BarcodeFormat** (LabelRequest XML) Obsolete: The Endicia Label Server will choose the proper barcode format based on the label type. ▪ **InsuredMail** (LabelRequest XML, PostageRateRequest XML) Added new value: UspsOnline ▪ **ReplyPostage** (LabelRequest XML) Updated: Enabled support for additional label types. ▪ **Services** (LabelRequest XML, PostageRateRequest XML) Added new attribute: ReturnReceipt. ▪ Added new error codes: 449 ExtraServiceNotAvailable

Date	Change
2008-03-20 Ver 3.3.3001	➢ Version 3.3 ▪ **NoHolidayDelivery** (LabelRequest XML) Obsolete. ▪ **NoWeekendDelivery** (LabelRequest XML) Updated: Endorsement will show as No Delivery Saturday. ▪ **ServiceLevel** (LabelRequest XML, PostageRateRequest XML) New: To set service level for Express Mail. ▪ **ShipDate, ShipTime** (LabelRequest XML, PostageRateRequest XML) New: To specify ship date and time for Express Mail Sunday/Holiday Delivery. ▪ Added new error codes: 2014 AccountPendingClose 2015 AccountIsClosed 11110 FailedRetrievingEMCommitments 11111 NotEligibleForSundayHolidayDelivery 11112 ServiceCommitmentsNotFound 11113 ServiceCommitmentNotOnSundayHoliday 11114 CannotMeetServiceCommitmentCutOffTime 11115 NoSundayHolidayDeliveryToDestination
2008-03-01 Ver 3.2.2982	➢ Version 3.2 Added support for the new USPS Priority Mail Large Flat Rate Box. ▪ **MailpieceShape** (LabelRequest XML, PostageRateRequest XML) Added new value: LargeFlatRateBox
2008-02-25	▪ **SCANRequest, CarrierPickupRequest, CarrierPickupCancel, CarrierPickupChange, MultiLocationCarrierPickup, TransactionListingRequest, RefundRequest, StatusRequest** Added PassPhrase node**.** ▪ **SCANRequest** Added PDF as ImageFormat**.** ▪ **Get Transactions Listing API** Corrected method name in documentation.
2008-02-08 Ver 3.1.2960	▪ As required by USPS regulations, the postage indicium in the label image returned in the LabelRequestResponse XML will contain the weight and zone information for all mailpieces which are classified as packages. ▪ **ImageResolution** (LabelRequest XML) Enabled support for 203 dpi for all image formats.

2008-01-08 Ver 3.1.2929	▪ **InsuredMail** (LabelRequest XML, PostageRateRequest XML) Enabled Endicia insurance for International mail. ▪ **InsuredValue** (LabelRequest XML, PostageRateRequest XML) New: To specify insured value. ***Backward compatibility note:*** When insurance is requested and InsuredValue is not supplied, it will be copied from the Value element. ▪ **LabelSize** (LabelRequest XML) Added new value: Mailer7x5 (for LabelType of DestinationConfirm) ▪ **Value** (LabelRequest XML, PostageRateRequest XML) Updated description.
2008-01-02	▪ **UserSignup** Changed length of CheckingAccountNumber and CheckingAccountRoutingNumber Nodes
2008-01-02	▪ **SCANRequest, CarrierPickupRequest, CarrierPickupCancel, CarrierPickupChange, MultiLocationCarrierPickup, TransactionListingRequest, RefundRequest, StatusRequest** Updated length of WebPassword node.
2007-12-13	▪ **UserSignup** Updated documentation to include remaining error messages Updated documentation to include notes about formatting of nodes. ▪ **SCANRequest, CarrierPickupRequest, CarrierPickupCancel, CarrierPickupChange, MultiLocationCarrierPickup** Added API
2007-11-09 Ver 3.1.2869	▪ **MailpieceShape** (LabelRequest XML, PostageRateRequest XML) USPS does not allow Delivery or Signature Confirmation service for First-Class Mail letter-size and flat-size mailpieces or Package Services flat-size mailpieces. To enforce this rule, the Endicia Label Server will now return one of the following error codes for an ineligible MailpieceShape and Confirmation Service combination: DeliveryConfirmationNotAvailable SignatureConfirmationNotAvailable
2007-10-18 Ver 3.1.2847	▪ **BarcodeFormat** (LabelRequest XML) Updated description. ▪ **ImageRotation (**LabelRequest XML) Enabled support for all rotation values. ▪ **MailClass** (LabelRequest XML, PostageRateRequest XML) Added new value: BoundPrintedMatter

2007-09-21	▪ **RefundRequest** Fixed documentation to show that the ErrMsg is returned regardless. Fixed documentation to show that it returns YES and NO, instead of True and False. ▪ **TransactionsListingsRequest** Added From Address info to documentation.
2007-08-23 Ver 3.1.2791	▪ **LabelSize** (LabelRequest XML) Added new value: 6x4 (for LabelType of Default)
2007-08-02 Ver 3.1.2770	➢ <u>Version 3.1</u> ▪ Added support for label output in ZPL II Programming Language: ▪ **ImageFormat** (LabelRequest XML). Added new value: ZPLII ▪ **LabelSize** (LabelRequest XML) Added new value for use with CertifiedMail label type: 8X3 ▪ **Results** (TransactionListingRequest) Added Results tag. ▪ **StartDate, StartTime, EndDate, and EndTime** (TransactionListingRequest) Made these tags optional. ▪ **TransactionListingRequest** Re-ordered output to be soonest transaction to oldest. Fixed title for status code definitions. ▪ **UserSignup** Added error messages. ▪ **UserSignup, RefundRequest, StatusRequest, GetTransactionsListing** Added <Test /> node to API Documentation.
2007-05-10 Ver 3.0.2686	▪ **ImageFormat** (LabelRequest XML). Enabled EPL2 for creating Priority Mail International, Flat Rate Envelope labels.

2007-05-03 Ver 3.0.2679	➢ Version 3.0 Added support for the new 2007 USPS domestic and international rates. ▪ **CustomsCertify, CustomsSigner** (LabelRequest XML) New: To allow for electronic signing of international customs form.. ▪ **FlatRate**[1] (LabelRequest XML, PostageRateRequest XML) Obsolete. ▪ **Label** (LabelRequestResponse XML) New: To return the label as separate images. ▪ **LabelType** (LabelRequest XML) Added new value: International ▪ **MailClass** (LabelRequest XML, PostageRateRequest XML) Added new values: ExpressMailInternational [3] PriorityMailInternational [3] Deleted values: Postcard (*replaced by First*). IntlGEM [2] IntlGPM [2] IntlAirParcel [2] ▪ **MailpieceDimensions**[4] (LabelRequest XML, PostageRateRequest XML) New: To set the dimensions of the mailpiece. ▪ **MailpieceShape**[5] (LabelRequest XML, PostageRateRequest XML) Added new values: Card (*replaces Postcard*) FlatRateBox FlatRateEnvelope IrregularParcel LargeParcel OversizedParcel Deleted value: Postcard (*replaced by Card*). ▪ **Oversize**[1] (LabelRequest XML) Obsolete. ▪ **PackageSize**[1] (LabelRequest XML, PostageRateRequest XML) Obsolete. ▪ **WeightOz** (LabelRequest XML, PostageRateRequest XML) Updated: Value allows for fractional ounces. **Backward compatibility notes:** 1. FlatRate, Oversize and PackageSize are obsolete and will no longer be supported after July 15, 2007. They have been superseded by MailpieceShape. If your code uses these elements, the Endicia Label Server Web Service will continue to map them to the

	appropriate MailpieceShape value. 2. IntlGEM, IntlGPM and IntlAirParcel will no longer be available starting May 14, 2007. If your code uses these mail classes, the Endicia Label Server Web Service will replace them with the new international mail classes: IntlGEM will be replaced with ExpressMailInternational; IntlGPM and IntlAirParcel will be replaced by PriorityMailInternational. 3. The label images for the new International mail classes have a different format. For example, Express Mail International has four parts compared to five in Global Express Mail. Priority Mail International Flat Rate Envelope has only one part compared to three for Global Priority Mail Flat Rate Envelope. 4. MailPieceDimensions is a new element that needs to be supplied for Priority Mail mailpieces going to Zones 5 – 8 and over 1 cubic foot to calculate the postage rate based on dimensional weight. Priority Mail Flat Rate Envelope and Priority Mail Flat Rate Box are not subject to this requirement. 5. MailpieceShape, if not supplied, will default to Parcel with the following exceptions: For First-Class DestinationConfirm and CertifiedMail labels, mailpiece shape will default to Letter if WeightOz is 3.5 oz or less; otherwise, it will default to Flat. For Priority Mail CertifiedMail labels, mailpiece shape will default to Flat.
2007-02-28 Ver 2.3.2615	▪ **LabelType** (LabelRequest XML) Destination Confirm labels require approval from USPS and Endicia. ▪ **PhoneNumberExtension** (UserSignupRequest XML) Updated: Fixed spelling, incorrectly spelled node still works. ▪ **StatusCode** (StatusRequestResponse XML) New: Added new node to the response. Updated: Fixed typographical errors in response definition. ▪ **GetTransactionsListing** (Endicia Label Server Web Service) Added new API to get all transactions matching the input.
2007-02-27 Ver 2.3.2614	▪ **RubberStamp1, RubberStamp2, RubberStamp3** (LabelRequest XML) Updated: Added Rubber Stamps to additional labels. ▪ **ToZIP4** (LabelRequest XML) Updated: Value is optional.
2007-02-21 Ver 2.3.2608	▪ **AutomationRate** (LabelRequest XML, PostageRateRequest XML) New: To use automation rates. ▪ **LabelSize** (LabelRequest XML) Added new value for use with DestinationConfirm label type: Dymo30384 ▪ **MailpieceShape** (LabelRequest XML, PostageRateRequest XML)

	New: To select shape of the mailpiece. ▪ **ReplyPostage (**LabelRequest XML) New: To print labels with reply postage. ▪ **ShowReturnAddress (**LabelRequest XML) New: To allow labels to print without the sender's address. ▪ **SortType** (LabelRequest XML, PostageRateRequest XML) New: To select sort level. ▪ **ValidateAddress (**LabelRequest XML) New: To bypass Dial-A-ZIP address validation.
2007-02-09 Ver 2.3.2596	▪ **ImageRotation (**LabelRequest XML) New: To allow rotation of label image. ▪ **LabelSize** (LabelRequest XML) Added new value for use with Default label type: DocTab
2007-01-30 Ver 2.3.2586	▪ **CalculatePostageRate, CalculatePostageRateXML** (Endicia Label Server Web Service) Added new Web methods to calculate postage rate. ▪ **ImageFormat** (LabelRequest XML). Removed support for BMP format. ▪ **LabelSize** (LabelRequest XML) Added new values for use with DestinationConfirm label type: EnvelopeSize10 Postcard ▪ **MailClass** (LabelRequest XML) Added new value: Postcard ▪ **RubberStamp1, RubberStamp2, RubberStamp3** (LabelRequest XML) New: To allow printing of user-supplied values. ▪ **Stealth** (LabelRequest XML) Updated: Defaults to TRUE. ▪ Sender's address is being validated; city, state and ZIP Code line is standardized.
2006-11-12 Ver 2.2.2507	▪ **GetAccountStatus, GetAccountStatusXML** (Endicia Label Server Web Service) Added new Web methods to get the status of an account.
2006-11-03 Ver 2.2.2498	▪ **FromCity, ToCity** (LabelRequest XML) City name can contain periods.
2006-10-29 Ver 2.2.2493	▪ **POZipCode (**LabelRequest XML) New: To specify mailing location. ▪ **FromName** (LabelRequest XML) Value is optional. ▪ **FromCompany** (LabelRequest XML)

	Value is required for Express Mail if FromName is omitted. ▪ **ToName** (LabelRequest XML) Value is optional. ▪ **ToCompany** (LabelRequest XML) Value is required for Express Mail if ToName is omitted. ▪ **RecreditAmount** (RecreditRequest XML) Added new predefined amounts: 7500, 10000, 20000.
2006-10-11 Ver 2.2.2482	▪ **ToPhone** (LabelRequest XML) Updated description.
2006-10-11 Ver 2.2.2475	▪ **RecreditAmount** (RecreditRequest XML) Field allows any amount of postage up to $99,999.99. ▪ Changed the Service Icon Text from "**1**" to "**FCM**" for First-Class Mail labels.
2006-09-20 Ver 2.2.2454	➢ <u>Version 2.2</u> ▪ Added support for customs forms: ▪ New in LabelRequest XML: **CustomsFormType** **CustomsFormImageFormat** **CustomsFormImageResolution** ▪ New in LabelRequestResponse XML: **CustomsForm** ▪ **LabelType** (LabelRequest XML) Added new value: CertifiedMail ▪ **LabelSize** (LabelRequest XML) Added new values for use with CertifiedMail label type: 4X6 7X4 Booklet EnvelopeSize10 ▪ **Base64LabelImageOptional** (LabelRequestResponse XML) Obsolete.
2006-08-27 Ver 2.1.2430	▪ Added support for Delivery Confirmation to APO/FPO addresses as well as selected Pacific Island ZIP Codes. ▪ Enabled Certified Mail and Electronic Return Receipt for Default label type.
2006-08-08 Ver 2.1.2411	▪ **LabelSize** (LabelRequest XML) Added new value: 6x4 (for LabelType of DestinationConfirm)
2006-07-24 Ver 2.1.2396	▪ **LabelType** (LabelRequest XML) Added new value: DestinationConfirm ▪ **LabelSize** (LabelRequest XML)

	Added new value for use with DestinationConfirm label type: 7X3
2006-07-19 Ver 2.1.2391	▪ **ContentsType** (LabelRequest XML) Added new value: ReturnedGoods ▪ **ContentsExplanation (**LabelRequest XML) New: To describe contents when ContentsType is set to OTHER. ▪ **BpodClientDunsNumber (**LabelRequest XML) New: To set DUNS number of partner.
2006-06-30 Ver 2.1.2372	▪ **FromCity, ToCity** (LabelRequest XML) City name can contain hyphens.
2006-06-28 Ver 2.1.2370	▪ **LabelType** (LabelRequest XML) Added new value: CertifiedMail
2006-05-23	▪ **MiddleInitial** (UserSignup XML) Changed Middle Initial to not required.
2006-05-17 Ver 2.1.2328	▪ **Services** (LabelRequest XML) Added new attribute: ElectronicReturnReceipt.
2006-05-16 Ver 2.1.2327	▪ **BarcodeFormat** (LabelRequest XML) Updated description. ▪ **ImageResolution (**LabelRequest XML) New: To allow selection of image resolution. ▪ **Services** (LabelRequest XML) Added new attribute: CertifiedMail. ▪ **TrackingType** (LabelRequest XML) Obsolete.
2006-04-06	▪ **PassPhrase** (LabelRequest XML, RecreditRequest XML, ChangePassPhraseRequest XML) Reduced size to 64 from 150. ▪ **NewPassPhrase** (ChangePassPhraseRequest XML) Reduced size to 64 from 150.
2006-03-22	➢ Version 2.1 ▪ Added support for label output in EPL2 Programming Language: ▪ **ImageFormat** (LabelRequest XML). Added new value: EPL2
2006-02-07	➢ Version 2.0 ▪ Added support for International Mail:

	▪ Updated in LabelRequest XML: **MailClass** **ToPostalCode** **ToCountry** ▪ New in LabelRequest XML: **ReturnToSender** **OriginCountry** **ContentsType** **CustomsDescription1, CustomsQuantity1, CustomsWeight1, CustomsValue1, CustomsCountry1** **CustomsDescription2, CustomsQuantity2, CustomsWeight2, CustomsValue2, CustomsCountry2** **CustomsDescription3, CustomsQuantity3, CustomsWeight3, CustomsValue3, CustomsCountry3** **CustomsDescription4, CustomsQuantity4, CustomsWeight4, CustomsValue4, CustomsCountry4** **CustomsDescription5, CustomsQuantity5, CustomsWeight5, CustomsValue5, CustomsCountry5** ▪ New in LabelRequestResponse XML: **CustomsNumber** ▪ Added support for PLANET Code labels (LabelRequest XML).
2006-01-11	▪ **IncludePostage** (LabelRequest XML) Added to allow PLANET Code labels to print without postage.
2005-12-12	➢ Version 1.2 ▪ Added support for current and new 2006 USPS domestic rates.
2005-08-15	▪ Bug fix (LabelRequest XML) Bug: Address verification would fail if the individual address elements of the Dial-A-ZIP XML were not URL encoded within the Endicia Label Server Web Service. Fix: URL encode the individual address elements of the Dial-A-ZIP XML.
2005-07-06	▪ **ToDeliveryPoint** (LabelRequest XML) Updated: Element is optional.
2005-07-05	▪ **LabelSize** (LabelRequest XML) Added new value: 4X4.5
2005-06-24	▪ LabelRequest XML The Endicia Label Server Web Service verifies and corrects, if necessary, the destination address. ▪ **MailClass** (LabelRequest XML) Added new value: LibraryMail ▪ **RecreditAmount** (RecreditRequest XML)

	Maximum amount of postage that can be added to the account has increased to $5,000.
2005-05-10	▪ **LabelSize** (LabelRequest XML) New: To select size of the label. ▪ **ToAddress1** (LabelRequest XML) Updated description.
2005-05-02	▪ Maximum insurable value for Endicia insurance has increased to $5,000.
2005-04-04	▪ **PackageSize** (LabelRequest XML) New: To set size of package to regular, large or oversize. ▪ **Oversize** (LabelRequest XML) Updated description.
2005-03-30	▪ Added support for the Stealth feature. ▪ Endicia Label Server Web Service calculates rates for "Local" zones properly. ▪ Returned label image in the LabelRequestResponse XML reflects the changes to address lines in the LabelRequest XML.
2005-03-24	▪ **ToName** (LabelRequest XML) Value is now optional except for Express Mail. ▪ **ToCompany** (LabelRequest XML) Value is required if ToAddress1 through ToAddress4 are omitted. ▪ **ToAddress1** (LabelRequest XML) Value is optional if ToCompany contains a value. ▪ **ToAddress5, ToAddress6, ReturnAddress5, ReturnAddress6** Obsolete. ***Backward compatibility note:*** If your code uses these obsolete tags, adjust accordingly. ▪ **ToName** **ToCompany** **ToAddress1, ToAddress2, ToAddress3, ToAddress4** **FromName** **FromCompany** **ReturnAddress1, ReturnAddress2, ReturnAddress3, ReturnAddress4** Reduced size to 47 characters from 150 characters. ***Backward compatibility note:*** If your code uses the extra characters for the name, company or delivery address lines, adjust accordingly.
2005-02-09	▪ **Machinable** (LabelRequest XML) New: To select machinable/non-machinable rate for Parcel Post. ▪ **Oversize** (LabelRequest XML) Updated to allow selection of oversize, non-machinable rate for Parcel Post.

2005-01-25	▪ **PIC** (LabelRequestResponse XML) Changed size of field to 22. The UCC/EAN Code128 barcode requires prepending of application identifier of "91" to the PIC number.
2005-01-04	▪ **BarcodeFormat** (LabelRequest XML) New: To support the UCC/EAN Code128 barcode format. ▪ **TrackingNumber** (LabelRequest XML) Changed size of PIC value to 22. The UCC/EAN Code128 barcode requires prepending of application identifier of "91" to the PIC number.

Appendix B. Error Codes

Error Code	Error Code String	Error Message
101	GenericError	**{Error Message}**
102	InternalEndiciaError	Internal Endicia Error: **{Error Message}**
307	InvalidOriginZipCode	Invalid origin ZIP Code.
308	InvalidDestinationZipCode	Invalid destination ZIP Code.
404	MailClassNotSupported	Mail class not supported (**{Mail Class}**).
405	InvalidMailClass	Invalid mail class (**{Mail Class}**).
406	MailpieceWeightMustBePositive	Mailpiece weight must be greater than zero.
407	MailpieceWeightExceedsMaximum	Weight (**{Weight}** ounces) exceeds maximum allowed for Mail Service (**{Weight}** ounces).
409	InvalidMailpieceValue	**{Description}** value is invalid.
410	MailpieceValueMustBePositive	**{Description}** value must be greater than $0.00.
411	MailpieceValueExceedsMaximum	Mailpiece value exceeds the **{Amount}** maximum allowed for **{Insurance Type}**.
412	MailpieceShapeNotSupported	Mailpiece shape not supported (**{Mailpiece Shape}**).
413	InvalidMailpieceShape	Invalid mailpiece shape (**{Mailpiece Shape}**).
414	ServiceLevelNotSupported	Service Level not supported (**{Service Level}**).
415	InvalidServiceLevel	Invalid Service Level (**{Service Level}**).
416	SortTypeNotSupported	Sort Type not supported (**{Sort Type}**).
417	InvalidSortType	Invalid Sort Type (**{Sort Type}**).
418	PricingNotOffered	**{Pricing}** pricing is not offered for **{Mail Service}**.
420	MailServiceNotAvailableToCountry	**{Mail Service}** is not available to **{Country Name}**.
421	InvalidCountry	**{Country}** is not a valid country.
422	OnlinePostageNotAllowed	Online postage is not allowed for **{Mail Service}** to **{Country Name}**.
423	LimitedMailServiceToCountry	**{Mail Service}** to **{Country Name}** is limited to **{allowed mail service(s)}**.
430	MailpieceDimensionsMustBePositive	Mailpiece length, width, and height must be greater than zero.
431	MailpieceDimensionsRequired	Mailpiece dimensions are required for **{Mail Class}**.
434	LengthPlusGirthExceedsMaximum	Mailpiece length plus girth of {**Dimension**}" exceeds maximum of {**Dimension**}" allowed by Mail Service.
436	InvalidEntryFacility	Invalid Entry Facility (**{Entry Facility}**).

Error Code	Error Code String	Error Message
437	EntryFacilityDoesNotServiceDesination	Entry Facility **{Entry Facility}** does not service destination ZIP code **{ZIP code}**.
440	DeliveryConfirmationNotAvailable	Delivery Confirmation is not available for **{Description}**.
441	SignatureConfirmationNotAvailable	Signature Confirmation is not available for **{Description}**.
448	InsuranceNotAvailableToCountry	**{Insurance Type}** is not available to **{Country Name}**.
449	ExtraServiceNotAvailable	**{Extra Service}** is not available for this mailpiece.
501	NoCubicPostageCalculator	Cubic postage calculator is not available for **{Mail Class} {Pricing}** pricing.
502	CubicPricingNotOffered	Cubic pricing is not offered for **{Mail Class} {Pricing}** pricing.
503	MailpieceCubicVolumeMustBePositive	Mailpiece cubic volume must be greater than zero.
504	MailpieceCubicVolumeExceedsMaximum	Cubic volume (**{Volume}** cu. ft.) exceeds maximum of **{Volume}** cu. ft. allowed for mail service.
1001	MissingOrInvalidXMLElement	Missing or invalid data element: **{Element Name}**
1002	ElementNotAllowed	Element not allowed: **{Element Name}**
1011	InvalidTrackingNumber	Service Type Code in Tracking Number must match Services requested for the package.
1020	InvalidRecreditAmount	RecreditAmount must be greater than zero and less than $100,000.
1050	NoCustomsItems	Customs Info does not contain any customs items.
1051	InvalidCustomsItemData	**{Element Name}** must be a positive numeric value **{Amount}** or less.
1052	TotalCustomsValueMustBePositive	Total customs value must be a positive numeric value 99999.99 or less.
1053	ContentsWeightExceedsPackageWeight	Total package weight must be equal to or greater than the sum of the item weights.
1054	ExceedsItemLimitForCustomsForm	**{Form Type}** is limited to only **{n}** items.
1055	ExceedsValueLimitForCustomsForm	Total customs value cannot exceed **{Amount}** for mailpiece using Customs Form **{Form Type}**.
1501	ExceedsLimitForAddressLines	Address lines exceed maximum allowed.
1505	MailClassNotAllowedForLabel	**{Mail Class}** is not allowed for label.
1101	AccountNotAuthorizedForLabel	Account #**{Account Number}** is not authorized for the requested label type..

Error Code	Error Code String	Error Message
2010	InvalidAccountStatus	Account #**{Account Number}** has an invalid status. Please contact the Endicia Label Server Support Team.
2011	AccountNeedsPassPhraseReset	Account #**{Account Number}** requires a Pass Phrase reset.
2012	AccountSuspendedBadLogins	Account #**{Account Number}** has been suspended due to multiple login attempts with an incorrect Pass Phrase. Please contact Endicia customer support at (650) 321-2640 x130 or LabelServer@endicia.com.
2013	AccountSuspendedForCause	Account #**{Account Number}** has been suspended. Please contact Endicia customer support at (650) 321-2640 x130 or LabelServer@endicia.com.
2014	AccountPendingClose	Account #**{Account Number}** is scheduled to be closed. Please contact Endicia customer support at (650) 321-2640 x130 or LabelServer@endicia.com.
2015	AccountIsClosed	Account #**{Account Number}** is closed. Please contact Endicia customer support at (650) 321-2640 x130 or LabelServer@endicia.com.
2020	AccountNotQualifiedForMailClass	Account #**{Account Number}** is not qualified for **{Mail Class}**.
2021	AccountNotQualifiedForFeature	Account #**{Account Number}** is not qualified for **{Feature Description}**.
3005	NameOrCompanyRequired	**{Sender's | Recipient's}** name or company required.
3010	InsuredValueExceedsEndiciaMaximum	Insured value exceeds the $10,000 maximum allowed by Endicia.
3011	InsuredValueMustBePositive	Insured value must be greater than $0.00.
3020	FailedValidatingShipFromAddress	Unable to validate Ship From address.
3021	FailedValidatingShipToAddress	Unable to validate Ship To address.
3023	ExpressMailNotAvailableToDestination	Express Mail Service is not available to the destination address.
3024	ServiceNotAvailableToDestination	One or more services are not available to the destination address.
3025	CustomsFormNotAllowedForMailpiece	Customs Form is not allowed for mailpiece.
4001	InvalidFieldInDialAZipResponse	The **{Field Name}** field in Dial-A-ZIP response is invalid.
11001	FailedRetrievingPICNumber	Unable to retrieve PIC number: **{Description}**
11002	PICNumberServiceError	Error returned from PIC Number Service.
11003	InvalidPICNumberReceived	Invalid PIC number received: **{Number}**

Error Code	Error Code String	Error Message
11101	FailedRetrievingExpressMailNumber	Unable to retrieve Express Mail number: **{Description}**
11102	ExpressMailAPIError	Error returned from Express Mail API: **{Description}**
11110	FailedRetrievingEMCommitments	Unable to retrieve Express Mail Service Commitments: **{Description}**
11111	NotEligibleForSundayHolidayDelivery	Mailpiece not eligible for Express Mail Sunday/Holiday Delivery Service.
11112	ServiceCommitmentsNotFound	There were no Express Mail Service Commitments for the mailpiece.
11113	ServiceCommitmentNotOnSundayHoliday	Express Mail Sunday/Holiday Delivery Service not allowed: Service Commitment does not fall on a Sunday or a Holiday.
11114	CannotMeetServiceCommitmentCutOffTime	Ship time is past the latest cut-off time to meet Express Mail Service Commitment.
11115	NoSundayHolidayDeliveryToDestination	Express Mail Sunday/Holiday Delivery Service is not available to ZIP Code **{Zip Code}**.
11201	FailedRetrievingPlanetCode	Unable to retrieve PLANET Code.
11202	PlanetCodeNumberServiceError	Error returned from PLANET Code Number Service: **{Description}**
11203	InvalidPlanetCodeNumberReceived	Invalid PLANET Code number received.
11301	FailedRetrievingLabelNumber	Unable to retrieve **{Mail Class/Extra Service}** number: **{Description}**
11302	LabelNumberServiceError	Error returned from Label Number Service: **{Description}**
11303	NoLabelNumberReceived	Label Number Service did not return a label number.
11304	InvalidLabelNumberReceived	Invalid label number received from Label Number Service: **{Label Number}**
11401	FailedRetrievingBarcodeNumber	Unable to retrieve **{Description}** barcode number.
11402	InternationalLabelAPIError	Error returned from International Label API: **{Description}**
11403	CustomsFormsAPIError	Error returned from Customs Forms API: **{Description}**
11451	FailedRetrievingAccountInformation	Unable to retrieve information for account #**{Account Number}**: **{Description}**
11452	AccountInformationServiceError	Error returned from Account Information Service for account #**{Account Number}**: **{Description}**
11453	InvalidValueinAccountInformation	Invalid **{Value Name}** value received for account #**{Account Number}**.
11501	FailedValidatingAddress	Unable to validate address: **{Description}**

Error Code	Error Code String	Error Message
11502	NonDeliverableAddress	The address is not a deliverable location according to the US Postal Service.
11503	UnknownDialAZipReturnCode	Dial-A-ZIP Server ({**Server Name**}) has returned an unknown return code of {**value**}.
11611	DialAZipInvalidZipCode	Dial-A-ZIP error: Invalid ZIP Code
11612	DialAZipInvalidStateCode	Dial-A-ZIP error: Invalid state code.
11613	DialAZipInvalidCity	Dial-A-ZIP error: Invalid city.
11621	DialAZipAddressNotFound	Dial-A-ZIP error: Address not found.
11622	DialAZipMultipleMatchesTooAmbiguious	Dial-A-ZIP error: Address is too ambiguous.
11625	DialAZipStreetAddressNotMatch	City, State and ZIP Code are valid, but street address is not a match.
12001	PurchaseTooSmall	Buy transaction must be greater than $10.00.
12002	IndiciumValueZero	Indicium value is zero.
12003	VpoTransactionFailed	VPO transaction failed (**{Status Codes}**): **{Description}**
12100	UnknownPostageTransactionError	The Postage Transaction has returned an unknown error.
12101	AccountNotFound	The Certified Intermediary's account number is invalid.
12103	InactiveAccount	The Certified Intermediary's account is not active.
12104	InsufficientFunds1	The Certified Intermediary's account does not have enough postage balance to support this transaction.
12115	IndiciumMaxExceeded	The indicium postage requested exceeds the account's limit
12116	InvalidCreditCard	Invalid credit card on account.
12117	InvalidCCExpiration	The account credit card's expiration date has passed
12118	InvalidPurchaseAmountLow	The purchase amount is too low.
12119	InvalidPurchaseAmountHigh	The purchase amount exceeds the account's purchase limit.
12120	CreditCardFailed	The credit card transaction failed.
12121	PurchaseUpdateFailed	The update to the account with the credit card information failed.
12124	DatabaseFailure	General database error
12125	InvalidRequest	The XML Request type is not recognized
12128	InvalidCCDeclined	The credit card purchase was declined by the customer's bank.

Error Code	Error Code String	Error Message
12129	InvalidCCReferred	The credit card purchase was “referred” by the customer’s bank. Contact the bank.
12130	Unavailable	The Postage Server is unavailable.
12131	InvalidCCAddress	The address on the credit card is invalid.
12132	CCUnavailable	The credit card processing is unavailable.
12133	Locked	The account is locked (possibly in use).
12134	AccountHoldFailed	The request to hold the account for a transaction failed.
12135	InternetAccessFailure	Internet access/communication failure.
12136	AccountIsBusy	Account is busy.
12201	XmlParseError	The XML request cannot be parsed for all required elements. The description provides details on the problem item.
12500	KeyMacCheckFailed	The key on the account has been tampered with and no transactions are allowed.
12502	NegativeDollarIndiciumRequest	Indicium requests must be positive.
12503	InsufficientFunds2	There is not enough money in the account to produce the indicium.
12505	AccountMacFailed	The account information has been tampered with and no transactions are allowed.
12507	IncorrectPassPhrase	The Certified Intermediary’s Pass Phrase is incorrect.
12514	ReplayDetected	The system has detected a duplicate request for postage.
12515	AccountMacCheckFailed	The account information has been tampered with and no transactions are allowed.
12525	AccountSuspended	The account has been suspended and no transactions are allowed.
13001	ConfirmReceiptFailed	Internal Error: Confirm Receipt transaction failed for label number {0}.
100002	InternalLabelServiceError	Internal Endicia Label Server Web Service Error: **{Error Message}**
100003	InvalidSoapRequestFormat	The format of the SOAP request is invalid.
100105	FailedCreatingPostageLabel	Failed to create postage label.
100106	FailedCreatingCustomsForm	Failed to create customs form.
100201	AccountNotQualifiedForPricing	Account #**{Account Number}** is not qualified for **{Mail Class} {Pricing}** pricing.

Appendix C. More Source Examples

This appendix contains various examples of source material to demonstrate how to perform certain actions, such as requesting a shipping label, including:

- Coldfusion Examples
- C# Examples
- Perl Examples
- PHP Examples
- Ruby Examples
- VisualBasic.NET Examples

Caution Statements

Note the following:

- No support or service of these samples is implied or acknowledged and use of it is completely at the user's discretion and responsibility.
- This section contains sample application programs in source language that illustrate programming techniques on various operating platforms. These examples have not been thoroughly tested under all conditions. Endicia, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
- The source code provided here are for sample use and are provided "as-is"; Endicia disclaims all express and implied warranties, including, the implied warranties of merchantability or fitness for a particular use. Endicia shall not be liable for any damages, including, direct, indirect, incidental, special or consequential damages for loss of profits, revenue, data or data use, incurred by you or any third party in connection with the use of these materials.
- This is a sample application and is provide "as is" and without warranty. The technical approach used in this sample implementation is one approach to achieve integration between the Endicia Label Server Web Service and a system. The techniques presented may not be appropriate for your system, for all scenarios, and within all development environments. Please make sure you consult the associated documentation of your development environment for further details.

Coldfusion Examples

This section shows one or more Coldfusion source examples:

- Request a First-Class Shipping Label: LabelServiceInvoke.cfm

Request a First-Class Shipping Label: LabelServiceInvoke.cfm

The following source example:

- Requests a First-Class shipping label, using the Request a Shipping Label API.
- Uses the ColdFusion development environment.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<cfsavecontent variable="soap_packet">
<cfoutput>
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetPostageLabel xmlns="www.envmgr.com/LabelService">
      <LabelRequest ImageFormat="JPEG" Test="YES">
            <RequesterID>TESTREQ</RequesterID>
            <AccountID>123456</AccountID>
            <PassPhrase>PASSWORD</PassPhrase>
            <MailClass>FIRST</MailClass>
            <WeightOz>1</WeightOz>
            <PartnerCustomerID>CUSTID</PartnerCustomerID>
            <PartnerTransactionID>1000</PartnerTransactionID>
            <ToName>Bob Jones</ToName>
            <ToAddress1>247 High St</ToAddress1>
            <ToCity>Palo Alto</ToCity>
            <ToState>CA</ToState>
            <ToPostalCode>94301</ToPostalCode>
            <ToZIP4>1041</ToZIP4>
            <ToDeliveryPoint>11</ToDeliveryPoint>
            <FromName>Steve Smith</FromName>
            <ReturnAddress1>247 High St</ReturnAddress1>
            <FromCity>Palo Alto</FromCity>
            <FromState>CA</FromState>
            <FromPostalCode>94301</FromPostalCode>
      </LabelRequest>
    </GetPostageLabel>
  </soap:Body>
</soap:Envelope>
</cfoutput>
</cfsavecontent>

<cfhttp url="https://www.envmgr.com/LabelService/EwsLabelService.asmx?WSDL"
method="POST">
 <cfhttpparam type="HEADER" name="Content-Type" value="text/xml; charset=utf-
8">
 <cfhttpparam type="HEADER" name="Host" value="www.envmgr.com">
```

```
 <cfhttpparam type="HEADER" name="SOAPAction"
value="www.envmgr.com/LabelService/GetPostageLabel">
 <cfhttpparam type="HEADER" name="Content-Length"
value="#len(trim(soap_packet))#">
 <cfhttpparam type="BODY" value="#trim(soap_packet)#">
</cfhttp>

<cfoutput>

<cfset myDoc =
xmlParse(Replacelist(cfhttp.fileContent,"soap:Body,soap:Envelope","body,envel
ope"))>

<cfset Image64 =
#myDoc.envelope.body.GetPostageLabelResponse.LabelRequestResponse.Base64Label
Image.XMLText#>

<cfset ImageData = #toBinary(Image64)#>

<CFSET timestamp =
"#DateFormat(Now(),'yyyymmdd')##TimeFormat(Now(),'HHmmssl')#">

<cffile action="write" file="#ExpandPath('.')#/tempimage#timestamp#.jpg"
output="#ImageData#" attributes="normal">

<img src="tempimage#timestamp#.jpg" />
</cfoutput>
```

C# Examples

This section shows one or more C# (C Sharp) source examples:

- Request a Refund: ConsumeRefundRequest.cs
- Request a List of Transactions: GetTransactionsListing.cs

Request a Refund: ConsumeRefundRequest.cs

The following source example:

- Requests a refund for a label using the Refund Web Service API.
- Uses the C# development environment.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
/*
 * Created by SharpDevelop.
 * User: troy
 * Date: 11/2/2006
 * Time: 11:43 AM
 *
 * To change this template use Tools | Options | Coding | Edit Standard
Headers.
 */
using System;
using System.Xml;
using System.Web.Services;
using System.Collections.Generic;
using System.Web;
using System.Net;
using System.IO;
using System.Text;

namespace ConsumeRefundRequest
{
      class MainClass
      {
            public static void Main(string[] args)
            {
                  Console.WriteLine("Start");

                  String strAccount = "";
                  String strWebPassword = "";
                  String strPICNumber = "";

                  String strXML = "<RefundRequest><AccountID>" + strAccount +
"</AccountID><WebPassword>" + strWebPassword +
"</WebPassword><RefundList><PICNumber>" + strPICNumber +
"</PICNumber></RefundList></RefundRequest>";

                  string postData = "method=RefundRequest&XMLInput=" +
strXML;
```

```
                HttpWebRequest request = (HttpWebRequest)
WebRequest.Create("https://www.endicia.com/ELS/ELSServices.cfc?wsdl");

                request.ContentType = "application/x-www-form-urlencoded";
                request.ContentLength = postData.Length;
                request.Method = "POST";
                request.AllowAutoRedirect = false;

                Stream requestStream = request.GetRequestStream();
                byte[] postBytes = Encoding.ASCII.GetBytes(postData);
                requestStream.Write(postBytes, 0, postBytes.Length);
                requestStream.Close();
                HttpWebResponse response = (HttpWebResponse)
request.GetResponse();

                Console.WriteLine(new
StreamReader(response.GetResponseStream()).ReadToEnd());
                Console.WriteLine("Headers:");
                Console.WriteLine(response.Headers.ToString());
            }
        }
}
```

Request a List of Transactions: GetTransactionsListing.cs

The following source example:

- Requests a list of transactions using the Get Transactions Listing API.
- Uses the C# development environment.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
/*
 * Created by SharpDevelop.
 * User: troy
 * Date: 3/6/2007
 * Time: 10:12 AM
 *
 * To change this template use Tools | Options | Coding | Edit Standard
Headers.
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Diagnostics;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml;
using System.Xml.Serialization;

namespace GetTransactionsListing
{
      class MainClass
      {
            public static void Main(string[] args)
            {
                  string AccountID  =     "";
                  string WebPassword      =     "";
                  string Connection =     "http://www.endicia.com/";
                  string xmlsend          =
      "<TransactionsListingRequest><AccountID>" + AccountID +
"</AccountID><WebPassword>" + WebPassword + "</WebPassword>" +

      "<StartDate>02/01/2007</StartDate><StartTime>00:00:00</StartTime><EndDa
te>02/07/2007</EndDate><EndTime>23:59:59</EndTime>" +

      "</TransactionsListingRequest>";

                  try
                  {
                        ELSServicesService request    =     new
ELSServicesService(Connection);
                        System.Xml.XmlNode[] answer   =     {(new
XmlDocument()).FirstChild,(new XmlDocument()).FirstChild};
```

```
                        XmlDocument send              =     new
XmlDocument();
                        send.InnerXml                 =     xmlsend;

                        answer                              =
      (System.Xml.XmlNode[])request.GetTransactionsListing((object)send.Inner
Xml);

                        Console.WriteLine(answer[2].OuterXml);
                  }
                  catch (Exception except)
                  {
                        Console.WriteLine("Cannot Load XML: " +
except.Message + "\n\n" + xmlsend);
                  }
            }
      }
}

[System.CodeDom.Compiler.GeneratedCodeAttribute("wsdl", "1.0.3705.0")]
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Web.Services.WebServiceBindingAttribute(Name="ELSServices.cfcSoapBind
ing", Namespace="http://ELS")]
[System.Xml.Serialization.SoapIncludeAttribute(typeof(QueryBean))]
[System.Xml.Serialization.SoapIncludeAttribute(typeof(CFCInvocationException)
)]

public partial class ELSServicesService :
System.Web.Services.Protocols.SoapHttpClientProtocol
{
    public ELSServicesService(string connection)
    {
        this.Url = connection + "/ELS/ELSServices.cfc";
    }

    public ELSServicesService()
    {
        this.Url = "https://www.endicia.com/ELS/ELSServices.cfc";
    }

    [System.Web.Services.Protocols.SoapRpcMethodAttribute("",
RequestNamespace="http://ELS", ResponseNamespace="http://ELS")]
    [return:
System.Xml.Serialization.SoapElementAttribute("GetTransactionsListingReturn")
]

    public object GetTransactionsListing(object XMLInput)
    {
        object[] results = this.Invoke("GetTransactionsListing", new object[]
{XMLInput});
        return ((object)(results[0]));
    }

    public System.IAsyncResult BeginGetTransactionsListing(object XMLInput,
System.AsyncCallback callback, object asyncState)
    {
```

```
        return this.BeginInvoke("GetTransactionsListing", new object[]
{XMLInput}, callback, asyncState);
    }

    public object EndGetTransactionsListing(System.IAsyncResult asyncResult)
    {
        object[] results = this.EndInvoke(asyncResult);
        return ((object)(results[0]));
    }
}

[System.CodeDom.Compiler.GeneratedCodeAttribute("wsdl", "1.0.3705.0")]
[System.SerializableAttribute()]
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Xml.Serialization.SoapTypeAttribute(Namespace="http://rpc.xml.coldfus
ion")]

public partial class QueryBean
{
    [System.Xml.Serialization.SoapElementAttribute(IsNullable=true)]

    public string[] columnList;

    [System.Xml.Serialization.SoapElementAttribute(IsNullable=true)]

    public object[] data;
}

[System.CodeDom.Compiler.GeneratedCodeAttribute("wsdl", "1.0.3705.0")]
[System.SerializableAttribute()]
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Xml.Serialization.SoapTypeAttribute(Namespace="http://rpc.xml.coldfus
ion")]

public partial class CFCInvocationException
{
}
```

Perl Examples

This section shows one or more Perl source examples:

- Request a First-Class Shipping Label: LabelRequestXML.pl

Request a First-Class Shipping Label: LabelRequestXML.pl

The following source example:

- Requests a First-Class shipping label using the GetPostageLabel Web method of the Endicia Label Server Web Service API.
- Uses Perl.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
#!/usr/bin/perl

use LWP::UserAgent;
use HTTP::Request;
use HTTP::Headers;
use XML::Simple;
use MIME::Base64;

my $objUserAgent = LWP::UserAgent->new;

$request = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<LabelRequest Test=\"YES\" LabelType=\"Default\" LabelSize=\"4X6\"
ImageFormat=\"GIF\">
 <RequesterID>abcd</RequesterID>
 <AccountID>123456</AccountID>
 <PassPhrase>samplePassPhrase</PassPhrase>
 <MailClass>FIRST</MailClass>
 <DateAdvance>0</DateAdvance>
 <WeightOz>1</WeightOz>
  <Stealth>FALSE</Stealth>
 <Services InsuredMail=\"OFF\" SignatureConfirmation=\"OFF\" />
  <Value>0</Value>
 <Description>Sample Label</Description>
 <PartnerCustomerID>12345ABCD</PartnerCustomerID>
 <PartnerTransactionID>6789EFGH</PartnerTransactionID>
 <ToName>Amine Khechfe</ToName>
 <ToCompany>Endicia</ToCompany>
 <ToAddress1>247 High Street</ToAddress1>
 <ToCity>Palo Alto</ToCity>
 <ToState>CA</ToState>
 <ToPostalCode>84301</ToPostalCode>
 <ToZIP4>0000</ToZIP4>
 <ToDeliveryPoint>00</ToDeliveryPoint>
 <ToPhone>8005763279</ToPhone>
 <FromName>John Doe</FromName>
 <ReturnAddress1>123 Main Street</ReturnAddress1>
 <FromCity>Boise</FromCity>
 <FromState>ID</FromState>
```

```
 <FromPostalCode>83702</FromPostalCode>
 <FromZIP4>7261</FromZIP4>
 <FromPhone>8005551212</FromPhone>
</LabelRequest>";

$contentlength = length($request);

$objHeader = HTTP::Headers->new(
 Host           => 'www.envmgr.com',
 Content_Type   => 'application/x-www-form-urlencoded',
 Content_Length => $contentlength
);

$request = "labelRequestXML=$request";

$objRequest = HTTP::Request-
>new("POST","https://www.envmgr.com/LabelService/EwsLabelService.asmx/GetPost
ageLabelXML",$objHeader,$request);

$objResponse = $objUserAgent->request($objRequest);

if (!$objResponse->is_error)
{
  print "sent=$request\n\n";
  $content = $objResponse->content;
  print "respond=$content\n\n";
}
else
{
  $sent = $objRequest->as_string;
  print "$sent\n\n";
  $error = $objResponse->error_as_HTML;
  print "error=$error\n\n";
  print "respond=$content\n\n";
}
```

PHP Examples

This section shows one or more PHP source examples:

- Request a Shipping Label: Endicia.php
- Request a Shipping Label: nuSoapLabelRequest.php
- Request a Shipping Label: PHP5LabelRequest.php
- Request a First-Class Shipping Label: cURLLabelRequest.php
- Request an International Label: PHP5LabelRequestInternationalCustoms.php
- Request a Shipping Rate: PHP5CalculatePostageRate.php
- Request a Refund: curlRefund.php
- Request a Refund Based On PICNumber: formRefund.php
- Request a Refund: nuSoapRefund.php
- Request a Label Refund: php5RefundRequest.php
- Request a New User Signup: nuSoapUserSignup.php
- Request the Status of a Shipment: nuSoapStatus.php
- Request a List of Transactions: GetTransactionsListing API
- Request a Mail Carrier Pickup: nuSoapMLCarrierPickup.php

Request a Shipping Label: Endicia.php

The following source example:

- Retrieves a postage label by opening a URL using fopen() and then making the request using the LabelRequest API.
- Uses PHP.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

$strGetLabelURL =
"https://www.envmgr.com/LabelService/EwsLabelService.asmx/GetPostageLabelXML"
;

$request = '
```

```
<LabelRequest ImageFormat="GIF" Test="YES">
     <RequesterID>abcd</RequesterID>
     <AccountID>123456</AccountID>
     <PassPhrase>samplePassPhrase</PassPhrase>
     <MailClass>FIRST</MailClass>
     <DateAdvance>0</DateAdvance>
     <WeightOz>1</WeightOz>
     <Stealth>FALSE</Stealth>
     <Services InsuredMail="OFF" SignatureConfirmation="OFF" />
     <Value>0</Value>
     <Description>Sample Label</Description>
     <PartnerCustomerID>12345ABCD</PartnerCustomerID>
     <PartnerTransactionID>6789EFGH</PartnerTransactionID>
     <ToName>Amine Khechfe</ToName>
     <ToCompany>Endicia</ToCompany>
     <ToAddress1>247 High Street</ToAddress1>
     <ToCity>Palo Alto</ToCity>
     <ToState>CA</ToState>
     <ToPostalCode>84301</ToPostalCode>
     <ToZIP4>0000</ToZIP4>
     <ToDeliveryPoint>00</ToDeliveryPoint>
     <ToPhone>8005763279</ToPhone>
     <FromName>John Doe</FromName>
     <ReturnAddress1>123 Main Street</ReturnAddress1>
     <FromCity>Boise</FromCity>
     <FromState>ID</FromState>
     <FromPostalCode>83702</FromPostalCode>
     <FromZIP4>7261</FromZIP4>
     <FromPhone>8005551212</FromPhone>
    </LabelRequest>
';

      $params = array('http' => array(
                  'method' => 'POST',
                  'content' => 'labelRequestXML='.$request,
                        'header' => 'Content-Type: application/x-www-form-
urlencoded'));

      $ctx = stream_context_create($params);

      $fp = fopen($strGetLabelURL, 'rb', false, $ctx);

            if (!$fp) {
                  print "Problem with $strGetLabelURL";
            }
            $response = stream_get_contents($fp);
            if ($response === false) {
                  print "Problem reading data from $url, $php_errormsg";
            }
print_r($response);
?>
```

Request a Shipping Label: nuSoapLabelRequest.php

The following source example:

- Uses SOAP to request a shipping label using the Request a Shipping Label method of the Endicia Label Server API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

require_once('nusoap.php');

$soapclient = new
soapclient('https://www.envmgr.com/LabelService/EwsLabelService.asmx?wsdl');

$request = "
<GetPostageLabel xmlns=\"www.envmgr.com/LabelService\">
<LabelRequest Test=\"YES\" LabelType=\"Default\" LabelSize=\"4X6\"
ImageFormat=\"GIF\">
 <RequesterID>abcd</RequesterID>
 <AccountID>123456</AccountID>
 <PassPhrase>samplePassPhrase</PassPhrase>
 <MailClass>FIRST</MailClass>
 <DateAdvance>0</DateAdvance>
 <WeightOz>1</WeightOz>
 <Stealth>FALSE</Stealth>
 <Services InsuredMail=\"OFF\" SignatureConfirmation=\"OFF\" />
 <Value>0</Value>
 <Description>Sample Label</Description>
 <PartnerCustomerID>12345ABCD</PartnerCustomerID>
 <PartnerTransactionID>6789EFGH</PartnerTransactionID>
 <ToName>Amine Khechfe</ToName>
 <ToCompany>Endicia</ToCompany>
 <ToAddress1>247 High Street</ToAddress1>
 <ToCity>Palo Alto</ToCity>
 <ToState>CA</ToState>
 <ToPostalCode>84301</ToPostalCode>
 <ToZIP4>0000</ToZIP4>
 <ToDeliveryPoint>00</ToDeliveryPoint>
 <ToPhone>8005763279</ToPhone>
 <FromName>John Doe</FromName>
 <ReturnAddress1>123 Main Street</ReturnAddress1>
 <FromCity>Boise</FromCity>
 <FromState>ID</FromState>
 <FromPostalCode>83702</FromPostalCode>
 <FromZIP4>7261</FromZIP4>
 <FromPhone>8005551212</FromPhone>
</LabelRequest>
</GetPostageLabel>
";
```

```
$soapclient-
>call('GetPostageLabel',$request,'http://schemas.xmlsoap.org/soap/envelope/',
'www.envmgr.com/LabelService/GetPostageLabel',false,null,'document','literal'
);

$XMLChar = array(">", "<");
$HTMLChar = array("&gt;","&lt;");

echo '<br /><hr /><br />Request: <pre>';
echo str_replace($XMLChar,$HTMLChar,$soapclient->request);
echo '</pre>';
echo '<br /><hr /><br />Response: <pre>';
echo str_replace($XMLChar,$HTMLChar,$soapclient->response);
echo '</pre>';
?>
```

Request a Shipping Label: PHP5LabelRequest.php

The following source example:

- Uses SOAP to request a shipping label using the Request a Shipping Label method of the Endicia Label Server API.
- Uses PHP5.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

$client = new
SoapClient("https://www.envmgr.com/LabelService/EwsLabelService.asmx?wsdl");

echo "<b>USING ARRAY</b><br /><br />";

$service_type = 'PriorityMailInternational';
$ounces     = '16';
$value = '500.00';
$to_fname = 'Troy Elseth';
$to_company = 'hofbrauhaus';
$to_addr1 = 'Hofbrauallee 1';
$to_city = 'Munich';
$to_state = ' ';
$to_zip5 = '81829';
$to_zip4 = '';
$to_country = 'Germany';
$to_phone = '6503212640';
$from_fname = 'Bob Smith';
$from_addr1 = '15442 36th Street NW';
$from_city = 'NEWFOLDEN';
$from_state = 'MN';
$from_zip5 = '56738';
$from_zip4 = '';
$from_phone = '2188743305';
$mime =     'GIF';

$data = array
(
      'LabelRequest' => array
      (
            'RequesterID' => 'THEREQUESTER',
            'AccountID' => '123456',
            'PassPhrase' => 'MyPassPhrase',
            'MailClass' => $service_type,
            'DateAdvance' => 0,
            'WeightOz' => $ounces ,
            'CostCenter' => 0,
            'Value' => $value,
            'Services' => array
            (
                  'CertifiedMail' => 'OFF',
```

```
                'DeliveryConfirmation' => 'ON',
                'ElectronicReturnReceipt' => 'OFF',
                'InsuredMail' => 'ENDICIA',
                'SignatureConfirmation' => 'OFF'
            ),
            'Description' => 'Sample Label',
            'PartnerCustomerID' => '12345ABCD',
            'PartnerTransactionID' => '6789EFGH',
            'OriginCountry' => 'United States',
            'ToName' =>  $to_fname,
            'ToCompany' => $to_company,
            'ToAddress1' => $to_addr1,
            'ToCity' => $to_city,
            'ToState' => $to_state,
            'ToPostalCode' => $to_zip5,
            'ToZIP4' => $to_zip4,
            'ToCountry' => $to_country,
            'ToPhone' => $to_phone,
            'FromName' => $from_fname,
            'ReturnAddress1' => $from_addr1,
            'FromCity' => $from_city,
            'FromState' => $from_state,
            'FromPostalCode' => $from_zip5,
            'FromZIP4' => $from_zip4,
            'FromPhone' => $from_phone,
            'CustomsQuantity1' => 0,
            'CustomsValue1' => 0,
            'CustomsWeight1' => 0,
            'CustomsQuantity2' => 0,
            'CustomsValue2' => 0,
            'CustomsWeight2' => 0,
            'CustomsQuantity3' => 0,
            'CustomsValue3' => 0,
            'CustomsWeight3' => 0,
            'CustomsQuantity4' => 0,
            'CustomsValue4' => 0,
            'CustomsWeight4' => 0,
            'CustomsQuantity5' => 0,
            'CustomsValue5' => 0,
            'CustomsWeight5' => 0,
            'Test' => 'YES',
            'LabelSize' => '4x6',
            'LabelType' => 'International',
            'ImageFormat' => $mime,
            'ImageResolution' => '203'
      )
);

echo "Sending Array:<br />";

print_r($data);
echo "<br /><br />Result Returned:<br />";

//var_dump($client->__getFunctions());

//var_dump($client->__getTypes());
```

```
$result = $client->GetPostageLabel($data);

//$image = $result->LabelRequestResponse->Base64LabelImage;
//header("Content-Type: image/gif");
//print_r(base64_decode($image));

print_r($result);

?>
```

Request a First-Class Shipping Label: cURLLabelRequest.php

The following source example:

- Uses URL to request a First-Class shipping label using the GetPostageLabel Web method of the Endicia Label Server Web Service API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

$request = "
<LabelRequest Test=\"YES\" LabelType=\"Default\" LabelSize=\"4X6\"
ImageFormat=\"GIF\">
 <RequesterID>abcd</RequesterID>
 <AccountID>123456</AccountID>
 <PassPhrase>samplePassPhrase</PassPhrase>
 <MailClass>FIRST</MailClass>
 <DateAdvance>0</DateAdvance>
 <WeightOz>1</WeightOz>
 <Stealth>FALSE</Stealth>
 <Services InsuredMail=\"OFF\" SignatureConfirmation=\"OFF\" />
 <Value>0</Value>
 <Description>Sample Label</Description>
 <PartnerCustomerID>12345ABCD</PartnerCustomerID>
 <PartnerTransactionID>6789EFGH</PartnerTransactionID>
 <ToName>Amine Khechfe</ToName>
 <ToCompany>Endicia</ToCompany>
 <ToAddress1>247 High Street</ToAddress1>
 <ToCity>Palo Alto</ToCity>
 <ToState>CA</ToState>
 <ToPostalCode>84301</ToPostalCode>
 <ToZIP4>0000</ToZIP4>
 <ToDeliveryPoint>00</ToDeliveryPoint>
 <ToPhone>8005763279</ToPhone>
 <FromName>John Doe</FromName>
 <ReturnAddress1>123 Main Street</ReturnAddress1>
 <FromCity>Boise</FromCity>
 <FromState>ID</FromState>
 <FromPostalCode>83702</FromPostalCode>
 <FromZIP4>7261</FromZIP4>
 <FromPhone>8005551212</FromPhone>
</LabelRequest>";

$postfields = 'labelRequestXML='.$request;

$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL,
'https://www.envmgr.com/LabelService/EwsLabelService.asmx/GetPostageLabelXML'
);
```

```
curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);

curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS, $postfields);

$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");

echo $curl_result;
?>
```

Request an International Label: PHP5LabelRequestInternationalCustoms.php

The following source example:

- Uses SOAP to request an international label using the Request a Shipping Label method of the Endicia Label Server API.

- Uses PHP5.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

$client = new
SoapClient("https://www.envmgr.com/LabelService/EwsLabelService.asmx?wsdl");

//echo "<b>USING ARRAY</b><br /><br />";

$service_type = 'PriorityMailInternational';
$ounces     = '16';
$value = '';
$to_fname = 'Troy Elseth';
$to_company = 'hofbrauhaus';
$to_addr1 = 'Hofbrauallee 1';
$to_city = 'Munich';
$to_state = ' ';
$to_zip5 = '81829';
$to_zip4 = '';
$to_country = 'Germany';
$to_phone = '6503212640';
$from_fname = 'Bob Smith';
$from_addr1 = '15442 36th Street NW';
$from_city = 'NEWFOLDEN';
$from_state = 'MN';
$from_zip5 = '56738';
$from_zip4 = '';
$from_phone = '2188743305';
$mime   =       'GIF';

$data = array
(
      'LabelRequest' => array
      (
            'RequesterID' => 'THEREQUESTER',
            'AccountID' => '123456',
            'PassPhrase' => 'MyPassPhrase',
            'MailClass' => $service_type,
            'DateAdvance' => 0,
            'WeightOz' => $ounces ,
            'CostCenter' => 0,
            'Value' => $value,
            'Services' => array
            (
                  'CertifiedMail' => 'OFF',
```

```
                    'DeliveryConfirmation' => 'ON',
                    'ElectronicReturnReceipt' => 'OFF',
                    'InsuredMail' => 'OFF',
                    'SignatureConfirmation' => 'OFF'
                ),
                'Description' => 'Sample Label',
                'PartnerCustomerID' => '12345ABCD',
                'PartnerTransactionID' => '6789EFGH',
                'OriginCountry' => 'United States',
                'ToName' =>  $to_fname,
                'ToCompany' => $to_company,
                'ToAddress1' => $to_addr1,
                'ToCity' => $to_city,
                'ToState' => $to_state,
                'ToPostalCode' => $to_zip5,
                'ToZIP4' => $to_zip4,
                'ToCountry' => $to_country,
                'ToPhone' => $to_phone,
                'FromName' => $from_fname,
                'ReturnAddress1' => $from_addr1,
                'FromCity' => $from_city,
                'FromState' => $from_state,
                'FromPostalCode' => $from_zip5,
                'FromZIP4' => $from_zip4,
                'FromPhone' => $from_phone,
                'CustomsQuantity1' => 0,
                'CustomsValue1' => 0,
                'CustomsWeight1' => 0,
                'CustomsQuantity2' => 0,
                'CustomsValue2' => 0,
                'CustomsWeight2' => 0,
                'CustomsQuantity3' => 0,
                'CustomsValue3' => 0,
                'CustomsWeight3' => 0,
                'CustomsQuantity4' => 0,
                'CustomsValue4' => 0,
                'CustomsWeight4' => 0,
                'CustomsQuantity5' => 0,
                'CustomsValue5' => 0,
                'CustomsWeight5' => 0,
                'Test' => 'YES',
                'LabelSize' => '4X6',
                'LabelType' => 'International',
                'ImageFormat' => $mime,
                'ImageResolution' => '150'
        )
);

//echo "Sending Array:<br />";

//print_r($data);
//echo "<br /><br />Result Returned:<br />";

//var_dump($client->__getFunctions());

//var_dump($client->__getTypes());
```

```
$result = $client->GetPostageLabel($data);

$image = $result->LabelRequestResponse->Label->Image[0]->_;

$file = './Temp/Label.gif';

$fh = fopen($file, "w+");
$file_contents = base64_decode($image);
fwrite($fh, $file_contents);
fclose($fh);

$image = $result->LabelRequestResponse->Label->Image[1]->_;

$file = './Temp/Label2.gif';

$fh = fopen($file, "w+");
$file_contents = base64_decode($image);
fwrite($fh, $file_contents);
fclose($fh);

$image = $result->LabelRequestResponse->Label->Image[2]->_;

$file = './Temp/Label3.gif';

$fh = fopen($file, "w+");
$file_contents = base64_decode($image);
fwrite($fh, $file_contents);
fclose($fh);

$image = $result->LabelRequestResponse->Label->Image[3]->_;

$file = './Temp/Label4.gif';

$fh = fopen($file, "w+");
$file_contents = base64_decode($image);
fwrite($fh, $file_contents);
fclose($fh);

//header("Content-Type: image/gif");
//print_r(base64_decode($image));

//print_r($result);

//print_r($image);

?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML lang=en XML:LANG="en" XMLNS="http://www.w3.org/1999/xhtml">
<HEAD>
<meta http-equiv="imagetoolbar" content="no" />
<TITLE>Endicia Label Service - Label.gif</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-1252">

</HEAD>
<BODY bgColor=#FFFFFF leftMargin=0 topMargin=0 MARGINWIDTH="0"
MARGINHEIGHT="0">
```

```
<TABLE>
<TR>
<TD><IMG SRC="./Temp/Label.gif" width="50%" height="50%"><br /></TD>
</TR>
<TR>
<TD><IMG SRC="./Temp/Label2.gif" width="50%" height="50%"><br /></TD>
</TR>
<TR>
<TD><IMG SRC="./Temp/Label3.gif" width="50%" height="50%"><br /></TD>
</TR>
<TR>
<TD><IMG SRC="./Temp/Label4.gif" width="50%" height="50%"><br /></TD>
</TR>
</TABLE>

</BODY>
</HTML>
```

Request a Shipping Rate: PHP5CalculatePostageRate.php

The following source example:

- Uses SOAP to request a shipping price using the Calculate Postage and Fees method of the Endicia Label Server API.
- Uses PHP5.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

$client = new
SoapClient("https://www.envmgr.com/LabelService/EwsLabelService.asmx?wsdl");

echo "<b>USING ARRAY</b><br /><br />";

$data = array
(
      'PostageRateRequest' => array
      (
            'RequesterID' => 'Tester1',
            'CertifiedIntermediary' => array
            (
                  'AccountID' => '123456',
                  'PassPhrase' => 'MyPassPhrase'
            ),
            'MailClass' => 'Priority',
            'WeightOz' => '54',
            'Value' => '100.00',
            'Services' => array
            (
                  '@attributes' => array
                  (
                        'DeliveryConfirmation' => 'TRUE',
                        'SignatureConfirmation' => 'OFF'
                  )
            ),
            'FromPostalCode' => '94301',
            'ToPostalCode' => '56738'
      )
);

echo "Sending Array:<br />";
print_r($data);
echo "<br /><br />Result Returned:<br />";

$result = $client->CalculatePostageRate($data);

print_r($result);

echo "<br /><br />Using the variables:<br />Status: " . $result-
>PostageRateResponse->Status . "<br />";
```

```
echo "Zone: " . $result->PostageRateResponse->Zone . "<br />";
echo "MailService: " . $result->PostageRateResponse->Postage->MailService .
"<br />";
echo "Rate: " . $result->PostageRateResponse->Postage->Rate . "<br /><br />";

?>
```

Request a Refund: curlRefund.php

The following source example:

- Uses cURL to request a refund for a shipping label using the Refund Web Service API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?

$xml = "";
$xml .= "<RefundRequest>\n";
$xml .= "<AccountID></AccountID>\n";
$xml .= "<WebPassword></WebPassword>\n";
$xml .= "<RefundList>\n";
$xml .= "<PICNumber>9101148008600086672136</PICNumber>\n";
$xml .= "</RefundList>\n";
$xml .= "</RefundRequest>\n";

$curl_handle = curl_init ();

curl_setopt ($curl_handle, CURLOPT_URL,
'http://www.endicia.com/ELS/ELSServices.cfc?wsdl');

$postfields = array(
      'method'   => 'RefundRequest',
      'XMLInput' => $xml
      );

curl_setopt ($curl_handle, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt ($curl_handle, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($curl_handle, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt ($curl_handle, CURLOPT_POST, 1);
curl_setopt ($curl_handle, CURLOPT_POSTFIELDS, $postfields);

$curl_result = curl_exec ($curl_handle) or die ("There has been a CURL_EXEC
error");

cu
```

Request a Refund Based On PICNumber: formRefund.php

The following source example:

- Uses SOAP to request a refund based on PICNumber using the Refund Web Service API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

require_once('nusoap.php');

$soapaddress = 'http://www.endicia.com/ELS/ELSServices.cfc?wsdl';
$soapclient = new soapclient($soapaddress);

$xml = "";
$xml = $xml."<RefundRequest>";
$xml = $xml."<AccountID></AccountID>";
$xml = $xml."<WebPassword></WebPassword>";
$xml = $xml."<RefundList>";
$xml = $xml."<PICNumber>9102148008600074274232</PICNumber>";
$xml = $xml."<PICNumber>9102148008600074417219</PICNumber>";
$xml = $xml."</RefundList>";
$xml = $xml."</RefundRequest>";

$params = array(
      'XMLInput' => $xml
            );

/*echo $xml;

$soapclient->call('RefundRequest',$params);

echo 'Response: <xmp>';
echo $soapclient->response;
echo '</xmp>';
*/
?>

<form name="MASTER" method="POST"
action="<?=$soapaddress?>&method=RefundRequest">
<input type="text" value="<?=$xml?>" name="XMLInput">
<input type="submit" name="submit" value="submit">
</form>
```

Request a Refund: nuSoapRefund.php

The following source example:

- Uses SOAP to request a refund using the Refund Web Service API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

require_once('nusoap.php');

$soapclient = new
soapclient('http://www.endicia.com/ELS/ELSServices.cfc?wsdl');

$xml = "";
$xml = $xml."<RefundRequest>";
$xml = $xml."<AccountID></AccountID>";
$xml = $xml."<WebPassword></WebPassword>";
$xml = $xml."<RefundList>";
$xml = $xml."<PICNumber>9102148008600074274232</PICNumber>";
$xml = $xml."<PICNumber>9102148008600074417219</PICNumber>";
$xml = $xml."</RefundList>";
$xml = $xml."</RefundRequest>";

$params = array(
      'XMLInput' => $xml
            );

echo $xml;

echo($soapclient->call('RefundRequest',$params));

echo 'Response: <xmp>';
echo $soapclient->response;
echo '</xmp>';

?>
```

Request the Status of a Shipment: nuSoapStatus.php

The following source example:

- Uses SOAP to request the status of a shipment using the Status Web Service API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

require_once('nusoap.php');

$soapclient = new
soapclient('http://192.168.1.5/ELS/ELSServicesString.cfc?wsdl');

$xml = "";
$xml = $xml."<StatusRequest>";
$xml = $xml."<AccountID></AccountID>";
$xml = $xml."<WebPassword></WebPassword>";
$xml = $xml."<StatusList>";
$xml = $xml."<PICNumber>9102148008600074274232</PICNumber>";
$xml = $xml."<PICNumber>9101148008600000041328</PICNumber>";
$xml = $xml."</StatusList>";
$xml = $xml."</StatusRequest>";

$params = array(
      'XMLInput' => $xml
            );

echo $xml;

echo($soapclient->call('StatusRequest',$params));

echo 'Response: <xmp>';
echo $soapclient->response;
echo '</xmp>';

?>
```

Request a List of Transactions: GetTransactionsListing.html

The following source example:

- Requests a list of transactions based on AccountID and WebPassword using the Get Transactions Listing API.
- Uses JavaScript.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<html>
<head>
<title>POST TEST</title>
<script type="text/javascript">
var AccountID = "";
var WebPassword = "";

function update(strURL,querystring,areaid)
{
      var xmlHttpReq = false;
    var self = this;

    // Mozilla/Safari
    if (window.XMLHttpRequest)
    {
        self.xmlHttpReq = new XMLHttpRequest();
    }
    // IE
    else if (window.ActiveXObject)
    {
        self.xmlHttpReq = new ActiveXObject("Microsoft.XMLHTTP");
    }
    self.xmlHttpReq.open('POST', strURL + '?' + querystring, true);
    self.xmlHttpReq.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
    self.xmlHttpReq.onreadystatechange = function()
    {
        if (self.xmlHttpReq.readyState == 4)
        {
            var output = self.xmlHttpReq.responseText;
            output = trim(output);
            eval("document.getElementById('" + areaid + "').innerHTML = \"" +
output + "\";");
        }
    }
    self.xmlHttpReq.send(querystring);
}

function trim(thestring)
{
      thestring = thestring.replace(/\n/g, "");
      thestring = thestring.replace(/^\s*|\s*$/g, "");
      thestring = thestring.replace(/\"/g,"'");
```

```
      thestring = thestring.replace(/</g,"&lt;");
      thestring = thestring.replace(/>/g,"&gt;");
      return thestring
}

function UpdateDiv()
{
      xmlsend           =     "<TransactionsListingRequest><AccountID>" +
AccountID + "</AccountID><WebPassword>" + WebPassword +
"</WebPassword><StartDate>02/01/2007</StartDate><StartTime>00:00:00</StartTim
e><EndDate>02/07/2007</EndDate><EndTime>23:59:59</EndTime></TransactionsListi
ngRequest>";
      querystring =     "&method=GetTransactionsListing&XMLInput=" + xmlsend;

      update("http://www.endicia.com/ELS/ELSServices.cfc?wsdl",querystring,"t
ester");
}
</script>

</head>
<body>
      <input type="button" name="button" value="Submit"
onClick="javascript:UpdateDiv();">
      <div id="tester" name="tester"
style="height:1024;width:768;border:solid 1px black;"></div>
</body>
</html>
```

Request a Mail Carrier Pickup: nuSoapMLCarrierPickup.php

The following source example:

- Uses SOAP to request a Mail Carrier Pickup using the Carrier Pickup Request API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

require_once('nusoap.php');

$soapclient = new
soapclient('http://65.203.54.167/ELS/ELSServices.cfc?wsdl');

$xml = "";
$xml = $xml."<MultiLocationCarrierPickup>";
$xml = $xml."<AccountID></AccountID>";
$xml = $xml."<WebPassword></WebPassword>";
$xml = $xml."</MultiLocationCarrierPickup>";

$params = array(
      'XMLInput' => $xml
            );

echo $xml;

echo($soapclient->call('MultiLocationCarrierPickup',$params));

echo '<br>Response: <xmp>';
echo $soapclient->response;
echo '</xmp>';

?>
```

Request a New User Signup: nuSoapUserSignup.php

The following source example:

- Uses SOAP to request a new user signup using the Account Signup API.
- Uses PHP4.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

require_once('nusoap.php');

$soapclient = new
soapclient('http://65.203.54.167/ELS/ELSServices.cfc?wsdl');

$xml = "";
$xml = $xml."<UserSignupRequest>";
$xml = $xml."<CompanyName>Example.COM</CompanyName>";
$xml = $xml."<FirstName>Jeff</FirstName>";
$xml = $xml."<MiddleInitial>M</MiddleInitial>";
$xml = $xml."<LastName>Nielsen</LastName>";
$xml = $xml."<EmailAddress>troy@endicia.com</EmailAddress>";
$xml = $xml."<EmailConfirm>eross@nexternal.com</EmailConfirm>";
$xml = $xml."<PhoneNumber>800-914-6161</PhoneNumber>";
$xml = $xml."<ICertify>Y</ICertify>";
$xml = $xml."<OverrideEmailCheck>Y</OverrideEmailCheck>";
$xml = $xml."<PhysicalAddress>785 Grand Ave Suite 216</PhysicalAddress>";
$xml = $xml."<PhysicalCity>Carlsbad</PhysicalCity>";
$xml = $xml."<PhysicalState>CA</PhysicalState>";
$xml = $xml."<PhysicalZipCode>92008</PhysicalZipCode>";
$xml = $xml."<WebPassword>password</WebPassword>";
$xml = $xml."<PassPhrase>passphrase</PassPhrase>";
$xml = $xml."<ChallengeQuestion>What's Up?</ChallengeQuestion>";
$xml = $xml."<ChallengeAnswer>Not much</ChallengeAnswer>";
$xml = $xml."<PartnerId>lnxt</PartnerId>";
$xml = $xml."<ProductType>LABELSERVER</ProductType>";
$xml = $xml."<CreditCardNumber>4111111111111111</CreditCardNumber>";
$xml = $xml."<CreditCardAddress>4811 Seascape Dr</CreditCardAddress>";
$xml = $xml."<CreditCardCity>Oceanside</CreditCardCity>";
$xml = $xml."<CreditCardState>CA</CreditCardState>";
$xml = $xml."<CreditCardZipCode>92057</CreditCardZipCode>";
$xml = $xml."<CreditCardType>V</CreditCardType>";
$xml = $xml."<CreditCardExpMonth>04</CreditCardExpMonth>";
$xml = $xml."<CreditCardExpYear>2006</CreditCardExpYear>";
$xml = $xml."<PaymentType>CC</PaymentType>";
$xml = $xml."</UserSignupRequest>";

$params = array(
      'XMLInput' => $xml
            );

$soapclient->call('UserSignup',$params);
```

```
echo 'Response:<br />';
echo $soapclient->response;
?>
```

Request a Label Refund: php5RefundRequest.php

The following source example:

- Uses SOAP to request a label refund using theRefundRequest API.
- Uses PHP5.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
<?php

$xml = "";
$xml .= "<RefundRequest>";
$xml .= "<AccountID>500000</AccountID>";
$xml .= "<WebPassword>THETESTPHRASE</WebPassword>";
$xml .= "<Test>Y</Test>";
$xml .= "<RefundList>";
$xml .= "<PICNumber>9101148008600086672136</PICNumber>";
$xml .= "</RefundList>";
$xml .= "</RefundRequest>";

$client = new SoapClient("https://www.endicia.com/ELS/ELSServices.cfc?wsdl");
//echo "<br /><br />";
//var_dump($client->__getFunctions());
//echo "<br /><br />";
//var_dump($client->__getTypes());

$result = $client->RefundRequest($xml);

echo "<br /><br />";

print_r($result);
?>
```

Ruby Examples

This section shows one or more Ruby source examples:

- Request a Shipping Label: RubyCURLLabelRequest.ruby

Request a Shipping Label: RubyCURLLabelRequest.ruby

The following source example:

- Uses cURL to request a shipping label using the Request a Shipping Label method of the Endicia Label Server API.
- Uses Ruby.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
#! /usr/bin/ruby -rrubygems
xml_request = "
<LabelRequest Test=\"YES\" LabelType=\"Default\" LabelSize=\"4X6\"
ImageFormat=\"GIF\">
 <RequesterID>abcd</RequesterID>
 <AccountID>123456</AccountID>
 <PassPhrase>samplePassPhrase</PassPhrase>
 <MailClass>FIRST</MailClass>
 <DateAdvance>0</DateAdvance>
 <WeightOz>1</WeightOz>
 <Stealth>FALSE</Stealth>
 <Services InsuredMail=\"OFF\" SignatureConfirmation=\"OFF\" />
 <Value>0</Value>
 <Description>Sample Label</Description>
 <PartnerCustomerID>12345ABCD</PartnerCustomerID>
 <PartnerTransactionID>6789EFGH</PartnerTransactionID>
 <ToName>Amine Khechfe</ToName>
 <ToCompany>Endicia</ToCompany>
 <ToAddress1>247 High Street</ToAddress1>
 <ToCity>Palo Alto</ToCity>
 <ToState>CA</ToState>
 <ToPostalCode>84301</ToPostalCode>
 <ToZIP4>0000</ToZIP4>
 <ToDeliveryPoint>00</ToDeliveryPoint>
 <ToPhone>8005763279</ToPhone>
 <FromName>John Doe</FromName>
 <ReturnAddress1>123 Main Street</ReturnAddress1>
 <FromCity>Boise</FromCity>
 <FromState>ID</FromState>
 <FromPostalCode>83702</FromPostalCode>
 <FromZIP4>7261</FromZIP4>
 <FromPhone>8005551212</FromPhone>
</LabelRequest>"

URL =
"https://www.envmgr.com/LabelService/EwsLabelService.asmx/GetPostageLabelXML"
```

```
require "curl"
c = Curl::Easy.http_post(URL, Curl::PostField.content('labelRequestXML',
xml_request))
c.follow_location = true
c.ssl_verify_host = false
puts c.body_str
```

VisualBasic.NET Examples

This section shows one or more VisualBasic.NET source examples:

- Requests a Shipping Label: LabelServiceClient.vb

Request a Shipping Label: LabelServiceClient.vb

The following source example:

- Requests a shipping label using the Request a Shipping Label method of the Endicia Label Server API.
- Uses VisualBasic.Net.

Important! Before proceeding, read and understand the **Caution Statements**.

Example

```
Imports LabelServiceClient.EwsLabelServiceProxy

Module LabelServiceClient

    Sub Main()

        Dim ewsLabelService As EwsLabelService
        Dim labelRequest As LabelRequest
        Dim labelRequestResponse As LabelRequestResponse

        labelRequest = New LabelRequest
        labelRequest.Services = New SpecialServices

        labelRequest.Test = "YES"
        labelRequest.ImageFormat = "GIF"
        labelRequest.LabelType = "DEFAULT"
        labelRequest.LabelSize = "4X6"
        labelRequest.RequesterID = "abcd"
        labelRequest.AccountID = "123456"
        labelRequest.PassPhrase = "samplePassPhrase"
        labelRequest.MailClass = "FIRST"
        labelRequest.DateAdvance = "0"
        labelRequest.WeightOz = Convert.ToUInt32(1)
        labelRequest.PackageSize = "REGULAR"
        labelRequest.Stealth = "FALSE"
        labelRequest.Services.InsuredMail = "OFF"
        labelRequest.Value = 0.0
        labelRequest.Description = "Sample Label"
        labelRequest.PartnerCustomerID = "12345ABCD"
        labelRequest.PartnerTransactionID = "6789EFGH"
        labelRequest.ToName = "Amine Khechfe"
        labelRequest.ToCompany = "Endicia"
        labelRequest.ToAddress1 = "247 High Street"
        labelRequest.ToCity = "Palo Alto"
        labelRequest.ToState = "CA"
        labelRequest.ToPostalCode = "84301"
```

```
        labelRequest.ToZIP4 = "0000"
        labelRequest.ToDeliveryPoint = "00"
        labelRequest.ToPhone = "8005763279"
        labelRequest.FromName = "John Doe"
        labelRequest.ReturnAddress1 = "123 Main Street"
        labelRequest.FromCity = "Boise"
        labelRequest.FromState = "ID"
        labelRequest.FromPostalCode = "83702"
        labelRequest.FromZIP4 = "7261"
        labelRequest.FromPhone = "8005551212"

        ewsLabelService = New EwsLabelService

        labelRequestResponse = ewsLabelService.GetPostageLabel(labelRequest)

        If (labelRequestResponse.Status = 0) Then
            System.Console.WriteLine("GetPostageLabel successful ... Label
image (in base64 string):")
            System.Console.WriteLine(labelRequestResponse.Base64LabelImage)
        Else
            System.Console.WriteLine("GetPostageLabel failed")
            System.Console.WriteLine("Status = " +
labelRequestResponse.Status.ToString())
        End If

    End Sub

End Module
```

Appendix D. Country Code Table

Use the table below for specifying the country code in the ToCountryCode element and the CountryOfOrigin element in CustomsInfo.

Country Code	Country
AF	Afghanistan
AX	Åland Islands
AL	Albania
DZ	Algeria
AS[1]	American Samoa
AD	Andorra
AO	Angola
AI	Anguilla
AQ[1]	Antarctica
AG	Antigua and Barbuda
AR	Argentina
AM	Armenia
AW	Aruba
AC[2]	Ascension Island
AU	Australia
AT	Austria
AZ	Azerbaijan
BS	Bahamas
BH	Bahrain
BD	Bangladesh
BB	Barbados
BY	Belarus
BE	Belgium
BZ	Belize
BJ	Benin
BM	Bermuda
BT	Bhutan
BO	Bolivia
BA	Bosnia and Herzegovina
BW	Botswana
BV[1]	Bouvet Island
BR	Brazil
IO[1]	British Indian Ocean Territory
BN	Brunei Darussalam
BG	Bulgaria
BF	Burkina Faso
BI	Burundi
KH	Cambodia

Country Code	Country
CM	Cameroon
CA	Canada
CV	Cape Verde
KY	Cayman Islands
CF	Central African Republic
TD	Chad
CL	Chile
CN	China
CX	Christmas Island
CC	Cocos (Keeling) Islands
CO	Colombia
KM	Comoros
CG	Congo
CD	Democratic Republic of the Congo
CK	Cook Islands
CR	Costa Rica
CI	Côte D'ivoire
HR	Croatia
CU	Cuba
CY	Cyprus
CZ	Czech Republic
DK	Denmark
DJ	Djibouti
DM	Dominica
DO	Dominican Republic
EC	Ecuador
EG	Egypt
SV	El Salvador
GQ	Equatorial Guinea
ER	Eritrea
EE	Estonia
ET	Ethiopia
FK	Falkland Islands
FO	Faroe Islands
FJ	Fiji
FI	Finland
FR	France
GF	French Guiana
PF	French Polynesia
TF[1]	French Southern Territories
GA	Gabon
GM	Gambia
GE	Georgia
DE	Germany

Country Code	Country
GH	Ghana
GI	Gibraltar
GR	Greece
GL	Greenland
GD	Grenada
GP	Guadeloupe
GU[1]	Guam
GT	Guatemala
GG	Guernsey
GN	Guinea
GW	Guinea-Bissau
GY	Guyana
HT	Haiti
HM[1]	Heard Island and Mcdonald Islands
VA	Vatican City
HN	Honduras
HK	Hong Kong
HU	Hungary
IS	Iceland
IN	India
ID	Indonesia
IR	Iran
IQ	Iraq
IE	Ireland
IM	Isle of Man
IL	Israel
IT	Italy
JM	Jamaica
JP	Japan
JE	Jersey
JO	Jordan
KZ	Kazakhstan
KE	Kenya
KI	Kiribati
KP	North Korea
KR	South Korea
XZ[3]	Kosovo
KW	Kuwait
KG	Kyrgyzstan
LA	Laos
LV	Latvia
LB	Lebanon
LS	Lesotho
LR	Liberia

Country Code	Country
LY	Libya
LI	Liechtenstein
LT	Lithuania
LU	Luxembourg
MO	Macao
MK	Macedonia
MG	Madagascar
MW	Malawi
MY	Malaysia
MV	Maldives
ML	Mali
MT	Malta
MH[1]	Marshall Islands
MQ	Martinique
MR	Mauritania
MU	Mauritius
YT	Mayotte
MX	Mexico
FM[1]	Federated States of Micronesia
MD	Republic of Moldova
MC	Monaco
MN	Mongolia
ME	Montenegro
MS	Montserrat
MA	Morocco
MZ	Mozambique
MM	Myanmar
NA	Namibia
NR	Nauru
NP	Nepal
NL	Netherlands
AN	Netherlands Antilles
NC	New Caledonia
NZ	New Zealand
NI	Nicaragua
NE	Niger
NG	Nigeria
NU	Niue
NF	Norfolk Island
MP[1]	Northern Mariana Islands
NO	Norway
OM	Oman
PK	Pakistan
PW[1]	Palau

Country Code	Country
PS[1]	Palestinian Territory, Occupied
PA	Panama
PG	Papua New Guinea
PY	Paraguay
PE	Peru
PH	Philippines
PN	Pitcairn
PL	Poland
PT[1]	Portugal
PR	Puerto Rico
QA	Qatar
RE	Reunion
RO	Romania
RU	Russia
RW	Rwanda
BL	Saint Barthélemy
SH	Saint Helena
KN	Saint Kitts and Nevis
LC	Saint Lucia
MF	Saint Martin
PM	Saint Pierre and Miquelon
VC	Saint Vincent and the Grenadines
WS	Samoa
SM	San Marino
ST	Sao Tome and Principe
SA	Saudi Arabia
SN	Senegal
RS	Serbia
SC	Seychelles
SL	Sierra Leone
SG	Singapore
SK	Slovakia
SI	Slovenia
SB	Solomon Islands
SO	Somalia
ZA	South Africa
GS	South Georgia
ES	Spain
LK	Sri Lanka
SD	Sudan
SR	Suriname
SJ[1]	Svalbard and Jan Mayen
SZ	Swaziland
SE	Sweden

Country Code	Country
CH	Switzerland
SY	Syrian Arab Republic
TW	Taiwan
TJ	Tajikistan
TZ	Tanzania
TH	Thailand
TL	Timor-Leste
TG	Togo
TK	Tokelau
TO	Tonga
TT	Trinidad and Tobago
TA[2]	Tristan da Cunha
TN	Tunisia
TR	Turkey
TM	Turkmenistan
TC	Turks and Caicos Islands
TV	Tuvalu
UG	Uganda
UA	Ukraine
AE	United Arab Emirates
GB	United Kingdom
US[1]	United States
UM[1]	United States Minor Outlying Islands
UY	Uruguay
UZ	Uzbekistan
VU	Vanuatu
VE	Venezuela
VN	Viet Nam
VG	British Virgin Islands
VI[1]	Virgin Islands
WF	Wallis and Futuna
EH[1]	Western Sahara
YE	Yemen
ZM	Zambia
ZW	Zimbabwe

1. USPS either does not offer service to that country, e.g. Antarctica (AQ), or the destination is considered domestic, e.g. Guam (GU). This Country Code value should not be used for ToCountryCode; however, it can be used for CountryOfOrigin in CustomsInfo.
2. Country Codes of AC, Ascension Island, and TA, Tristan de Cunha, have been exceptionally reserved by the ISO 3166/MA. Both are dependencies of Saint Helena (SH).
3. USPS has reserved a private Country Code of XZ for Kosovo to reflect its independence from the Republic of Serbia. For ISO, the Country Code will remain RS. Use either XZ or RS for Kosovo.

Appendix E. Country and Locality Listing

Use the table below for specifying the country or locality in the ToCountry element, the OriginCountry element, and the five customs item elements named CustomsCountry1 through CustomsCountry5.

Abu Dhabi	Belize	Chuuk
Admiralty Islands	Benin	Cocos Island
Afghanistan	Bermuda	Colombia
Aitutaki	Bhutan	Comoros
Ajman	Bismark Archipelago	Democratic Republic of the Congo
Aland Island	Bolivia	Republic of the Congo
Albania	Bonaire	Cook Islands
Alderney	Borabora	Corsica
Algeria	Borneo	Corisco Island
Alhucemas	Bosnia-Herzegovina	Costa Rica
Alofi Island	Botswana	Cote d Ivoire
Andaman Islands	Bougainville	Crete
Andorra	Bourbon	Croatia
Angola	Brazzaville	Cuba
Anguilla	Brazil	Cumino Island
Anjouan	British Guiana	Curacao
Annobon Island	British Honduras	Cyjrenaica
Antigua and Barbuda	British Virgin Islands	Cyprus
Argentina	Brunei Darussalam	Czech Republic
Armenia	Buka	Dahomey
Aruba	Bulgaria	Damao
Ascension	Burkina Faso	Danger Islands
Astypalaia	Burma	Denmark
Atafu	Burundi	Desirade Island
Atiu	Caicos Islands	Diu
Australia	Cambodia	Djibouti
Austria	Cameroon	Dodecanese Islands
Avarua	Canada	Doha
Azerbaijan	Canary Islands	Dominica
Azores	Canton Island	Dominican Republic
Bahamas	Cape Verde	Dubai
Bahrain	Cayman Islands	East Timor
Balearic Islands	Central African Rep	Ebeye
Baluchistan	Ceuta	Ecuador
Bangladesh	Ceylon	Egypt
Banks Island	Chad	Eire
Barbados	Chaferinas Islands	El Salvador
Barbuda	Chalki	Ellice Islands
Barthelemy	Channel Islands	Elobey Islands
Belarus	Chile	Enderbury Island
Belgium	China	England

Equatorial Guinea
Eritrea
Estonia
Ethiopia
Fakaofo
Falkland Islands
Fanning Island
Faroe Islands
Fernando Po
Fezzan
Fiji
Finland
Formosa
France
French Guiana
French Oceania
French Polynesia
French Somaliland
French Territory of the Afars and Issas
Friendly Islands
Fujairah
Futuna
Gabon
Gambia
Gambier
Republic of Georgia
Germany
Ghana
Gibraltar
Gilbert Islands
Goa
Gozo Island
Grand Comoro
Great Britain and Northern Ireland
Greece
Greenland
Grenada
Grenadines
Guadeloupe
Guatemala
Guernsey
Guinea
Guinea-Bissau
Guyana
Hainan Island

Haiti
Hashemite Kingdom
Hervey
Hivaoa
Holland
Honduras
Hong Kong
Huahine
Huan Island
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland
Irian Barat
Isle of Man
Isle of Pines
Israel
Italy
Jamaica
Japan
Jersey
Johore
Jordan
Kalimantan
Kalymnos
Kampuchea
Karpathos
Kassos
Kastellorizon
Kazakhstan
Kedah
Keeling Islands
Kelantan
Kenya
Kiribati
Kos
Kosovo
Kosrae
Kowloon
Kuwait
Kwajalein
Kyrgyzstan
Labrador

Labuan
Laos
Latvia
Lebanon
Leipsos
Leros
Les Saints Island
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Lord Howe Island
Loyalty Islands
Luxembourg
Macao
Macau
Macedonia
Madagascar
Madeira Islands
Majuro
Malacca
Malagasy Republic
Malawi
Malaya
Malaysia
Maldives
Mali
Malta
Manahiki
Manchuria
Mangaia
Manuai
Marie Galante
Marquesas Islands
Martinique
Mauke
Mauritania
Mauritius
Mayotte
Melilla
Mexico
Miquelon
Mitiaro
Moheli
Moldova

Monaco
Mongolia
Montenegro
Montserrat
Moorea
Morocco
Mozambique
Muscat
Myanmar
Namibia
Nansil Islands
Nauru
Negri Sembilan
Nepal
Netherlands
Netherlands Antilles
Netherlands West Indies
Nevis
New Britain
New Caledonia
New Hanover
New Hebrides
New Ireland
New South Wales
New Zealand
Newfoundland
Nicaragua
Niger
Nigeria
Nissiros
Niue
Norfolk Island
North Borneo
North Korea
Northern Ireland
Norway
Nukahiva
Nukunonu
Nyasaland
Ocean Island
Okinawa
Oman
Pahang
Pakistan
Palmerston
Panama

Papua New Guinea
Paraguay
Parry
Patmos
Pemba
Penang
Penghu Islands
Penon de Velez de la Gomera
Penrhyn
Perak
Perlis
Persia
Peru
Pescadores Islands
Petite Terre
Philippines
Pitcairn Island
Poland
Pohnpei
Portugal
Province Wellesley
Pukapuka
Qatar
Queensland
Quemoy
Raiatea
Rakaanga
Rapa
Rarotonga
Ras al Kaimah
Redonda
Reunion
Rhodesia
Rio Muni
Rodos
Rodrigues
Romania
Russia
Rwanda
Saba
Sabah
Saint Bartholomew
Saint Christopher
Saint Eustatius
Saint Helena
Saint Kitts

Saint Lucia
Saint Maarten
Saint Martin
Saint Pierre and Miquelon
Saint Vincent and the Grenadines
Salvador
San Marino
Santa Cruz Islands
Sao Tome and Principe
Sarawak
Sark
Saudi Arabia
Savage Island
British Savaii Island
Scotland
Selangor
Senegal
Serbia
Seychelles
Sharja
Shikoku
Siam
Sierra Leone
Sikkim
Singapore
Slovak Republic
Slovakia
Slovenia
Society Islands
Solomon Islands
Somali Democratic Republic
Somalia
Somaliland
South Africa
South Australia
South Georgia
South Korea
South-West Africa
Spain
Spitzbergen
Sri Lanka
Ste Marie de Madagascar
Sudan
Suriname
Suwarrow Islands
Swan Islands

Swaziland
Sweden
Switzerland
Symi
Syria
Syrian Arab Republic
Tahaa
Tahiti
Taiwan
Tajikistan
Tanzania
Tasmania
Tchad
Thailand
Thursday Island
Tibet
Tilos
Timor
Tobago
Togo
Tokelau Group
Tonga
Tongareva

Tori Shima
Torres Island
Trans-Jordan
Transkei
Trengganu
Trinidad and Tobago
Tripolitania
Tristan da Cunha
Trucial States
Truk
Tuamotou
Tubuai
Tunisia
Turkey
Turkmenistan
Turks and Caicos Islands
Tuvalu
Uganda
Umm Said
Umm al Quaiwain
Ukraine
Union Group
United Arab Emirates

United States
Upolu Island
Uruguay
Uzbekistan
Vanuatu
Vatican City
Venezuela
Victoria
Vietnam
Wales
Wallis and Futuna Islands
Wellesley Province
West New Guinea
Western Australia
Western Samoa
Yap
Yemen
Zafarani Islands
Zambia
Zanzibar
Zimbabwe

GLOSSARY

Account ID

A unique *Account ID* is assigned by Endicia when you sign up for the Endicia Label Server Web Service. The *Account ID* is associated with an Endicia postage account and holds information about postage transactions as well as available balance for printing postage.

When using the Test Server, you can use any 6 digit number for the Account ID. When on the production server, you will be supplied with an Endicia Label Server Account ID.

Girth

The distance around the thickest part of an object.

Pass Phrase

A very long and complex password. During the new account sign-up process, you will be asked to select a temporary *Pass Phrase* for the account. A Pass Phrase must be at least 5 characters long with a maximum of 64 characters. For added security, the Pass Phrase should be at least 10 characters long and include more than one word, use at least one uppercase and lowercase letter, one number and one non-text character, for example, punctuation.

Production Server

The *Production Server* contains the live version of the Endicia Label Server Web Service that generates shipping labels with real postage.

Requester ID

A *Requester ID* (also known as Partner ID or Partner Code) uniquely identifies the system making the request. Endicia assigns this ID during the new account sign-up process.

Test Server

The *Test Server* contains a test version of the Endicia Label Server Web Service that only generates "test" shipping labels with VOID postage. This allows you to develop and test your application's calls to the Endicia Label Server Web Service methods.

Web Password

You use the Web Password to:

- Gain access to the endicia.com Web site.
- View account transactions.
- Change account information.

The Web Password does not enable you to buy and print postage.

Web Service

A *Web Service* is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web Service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Reference source: The W3C at http://www.w3.org/TR/ws-gloss/

INDEX

A

account number
  assigned, receipt of · 8
  e-mail message · 82
account number information, receipt of email message · 9
account status
  flag · 75
  getting · 72
accounts
  about setting up · 8
  Active status · 72
  change Pass Phrases for · 51
  confirmation by e-mail message · 10
  creation through API · 8
  deciding how many · 9
  how many · 8
  ID · 219
  limits to number of · 8
  maintain multiple · 77, 91, 97, 104
  maintain security model of creating · 82
  payments made through · 7
  production, sign up for · 16
  reactivate · 140
  request status of · 72
  requirements to set up · 7
  resetting Pass Phrase · 9
  sign up · 11
  sign up · 82
  status flag · 48
  using single or multiple · 8
  Web Service to sign up · 11
AccountStatusRequest
  about · 72
  XML elements, table of · 74
  XML syntax · 73
  XML tags for · 73
  XMLexample of · 74
AccountStatusResponse
  XML elements, table of · 74
  XML syntax · 74
  XML tags for · 74
  XML, example of · 75
Active status, accounts with · 72, 75
address validation
  API · 132
  bypass · 25, 156
  Dial-A-ZIP · 132
  engine, rejection by · 141
addresses
  ambiguous, Dial-A-ZIP · 167
  APO/FPO · 136
  not a match, Dial-A-ZIP · 167
  not found, APO/FPO · 141
  not found, Dial-A-ZIP · 167
  sender's country · 34
  sender's postal code · 34
  unable to validate · 165
  validate · 132
  validate all · 25
agreement, Exhibit B · 140
API
  Buying Postage · 46
  Calculating Postage and Fees · 54, 64
  Carrier Pickup · 109
  Carrier Pickup Change · 123
  CarrierPickupRequest · 111
  Changing the Pass Phrase · 51
  creating accounts through · 8
  Getting Account Status · 72
  Multiple Location Carrier Pickup API · 109
  Requesting a Shipping Label · 17
APO/FPO
  address not found · 141
  addresses · 136
  addresses, confirmation restrictions · 136
  customs form type · 29
  customs forms · 141
automated carrier pickup · 136

B

bulk acceptance scans · 136
Buying Postage API · 46

C

C Sharp · *See* C#.
C# (C Sharp) examples
  ConsumeRefundRequest.cs · 173
  list of · 173
  Request a Refund · 173
CalculatePostageRate, to calculate postage · 54, 64
CalculatePostageRateXML, HTTP POST request for · 54, 64
Calculating Postage and Fees API · 54, 64
Canceling a Carrier Pickup Request API · 117
carrier pickup
  automated · 136
  cancel request · 117
  changing · 121
  PickupList container node · 123
  request a mail · 201
  requesting · 109
  schedule multiple locations · 127
  scheduling · 111
Carrier Pickup API · 109
Carrier Pickup Change API · 123
Carrier Pickup Request · *See also* CarrierPickupRequest.

CarrierPickupCancel
about · 117
error codes · 120
XML elements, table of · 117
XML, example of · 118
CarrierPickupCancelResponse
XML elements, table of · 118
XML syntax · 118
XML tags for · 118
XML, example of · 119
CarrierPickupChange
about · 121
error codes · 125
XML elements · 121
XML, example · 123
CarrierPickupChangeResponse
XML · 124
XML elements, table of · 118
XML elements, table of · 123
XML syntax · 123
XML tags · 123, 124
CarrierPickupRequest
about · 111
error codes · 116, 130
table of · 111
XML, example of · 113
CarrierPickupResponse
XML elements, table of · 113
XML tags · 113
XML tags for · 127
XML, example of · 114
caution statements, source examples · 170
ChangePassPhraseRequest
about · 51
XML elements, table of · 51
XML syntax · 51
XML tags for · 51
XML, example · 52
ChangePassPhraseRequestResponse
XML elements · 53
XML syntax · 53
XML tags for · 53
XML, example of · 53
changes
current · 12
history · 145
Changing a Carrier Pickup Request API · 121
Changing the Pass Phrase API · 51
city, invalid, Dial-A-ZIP · 167
Coldfusion
Request a First-Class Shipping Label · 171
source examples · 171
confirmation
delivery · 7, 26, 57, 66
delivery, fee · 44, 61, 69
delivery, not available · 164
eligible matter for delivery or signature · 137
e-mail address · 88
e-mail message for account · 10
linked to shipments · 136
number · 86
number for pickup · 118, 119
number of pickup · 128
number to change pickup · 123
number, change or cancel pickup · 109
number, missed pickup · 129
number, pickup · 114
number, to cancel or change pickup · 124
requirements for refunds · 140
restrictions for APO/FPO addresses · 136
signature · 7, 27, 28, 57, 66, 67
signature, fee · 44, 62, 69
signature, not available · 164
ConsumeRefundRequest.cs, source example · 173
country and locality listing · 215
cURLLabelRequest.php, source example · 187
curlRefund.php, source example · 195
current changes · 12
customs forms, tracking · 7

D

delivery
confirmation · 57, 66
confirmation fee · 44, 61, 69
confirmation not available · 164
delivery confirmation · *See also* confirmation.
eligible matter for · 137
Development Mode · 9
Dial-A-ZIP
address not found · 167
address validation · 132
ambiguous address · 167
invalid response · 166
invalid state code · 167
invalid ZIP Code · 167
street address not a match · 167
unknown return code · 167

E

elements
LabelRequest XML syntax · 17
LabelRequestResponse XML syntax · 42
email
sample label · 9
e-mail
for account number, receipt of · 8
Label Server Agreement · 9
message, Welcome · 6
e-mail
receipt of account number information · 9
e-mail
confirmation, for account · 10

e-mail
contacting support · 12
e-mail
message, Welcome · 77
e-mail
message, Welcome · 82
e-mail
account number · 82
e-mail
address, confirmation · 88
e-mail
message, Welcome · 91
e-mail
message, Welcome · 97
e-mail
message, Welcome · 109
e-mail
send sample label · 140
e-mail
signed agreement, Exhibit B · 140
e-mail
obtain support through · 140
e-mail
notifications, customize · 141
Endicia Label Server
accounts specific to · 6
team, contact · 12
Web service · 15
Endicia Label Server Web Service
accessing, through SSL/HTTPS · 137
first-time use · 16
instantiate · 6
Endicia Printable Postage XML · 136
Endicia.php, source example · 180
error codes
11110 · 138
11111 · 137
11112 · 137
11113 · 138
11114 · 138
11115 · 138
CarrierPickupCancel Error Codes · 120
CarrierPickupChange · 125
CarrierPickupRequest · 116
CarrierPickupRequest error codes · 130
GetTransactionsListing Response Error Codes · 103
list of · 163
RefundRequest · 80
SCANRequest · 108
Status element · 42, 48, 53, 60, 68, 75
StatusRequest · 96
USPS · 113, 116, 119, 120, 124, 126, 131
examples, source
C# · 173
Coldfusion · 171
cURLLabelRequest.php · 187
curlRefund.php · 195
Endicia.php · 180
formRefund.php · 196
GetTransactionsListing · 199
GetTransactionsListing.cs · 175
LabelRequestXML.pl · 178
nuSoapLabelRequest.php · 182
nuSoapMLCarrierPickup.php · 201
nuSoapRefund.php · 197
nuSoapStatus.php · 198
nuSoapUserSignup.php · 202
Perl · 178
PHP · 180
PHP5CalculatePostageRate.php · 193
PHP5LabelRequest.php · 184
PHP5LabelRequestInternationalCustoms.php · 189
php5RefundRequest.php · 204
Request a First-Class Shipping Label · 187
Request a List of Transactions · 175, 199
Request a Refund · 195, 197
Request a Refund Based On PICNumber · 196
Request a Shipping Label · 180, 184, 205, 207
Request a Shipping Rate · 193
Request an International Label · 189
Request the Status of a Shipment · 198
Ruby · 205
RubyCURLLabelRequest.ruby · 205
VisualBasic.NET · 207
Express Mail
automated carrier pickup · 136
counted as Express · 109
International · 55, 144
International, customs forms numbers · 7
label limitation · 35
label not available for · 19
mail class · 22, 55
no weekend delivery · 25
phone number, required · 34
recipient's e-mail address · 36
recipient's company name · 34
requirement for pickup change · 125
requirement for pickup request · 116, 131
sender's company name · 33
service level · 23, 56
ship date · 32
shipment time · 32
Sunday and holiday service · 137
tracking · 7

F

FAQ · 140
form submissions · 80, 90, 95, 107, 114, 119, 124, 130
formRefund.php, source example · 196
Frequently Asked Questions · 140

G

GetPostageLabel, about · 17
Getting Account Status API · 72
getting started · 6
GetTransactionsListing
  about · 97
  source example · 199
  XML elements, table of · 98
  XML syntax · 97
  XML tags · 97
  XML, syntax example · 100
GetTransactionsListing Response
  definitions · 103
  error codes · 103
  XML elements, table of · 100
  XML, example · 102
GetTransactionsListing.cs, source example · 175
girth · 219
glossary, basic terms · 219

H

history, changes · 145
holiday service, Express Mail · 137
HTTP POST
  request and response · 16
  request for CalculatePostageRateXML · 54, 64

I

international
  customs forms for · 7
  customs signer, required · 33
  Express Mail · 144
  label type · 20
  label, request for · 189
  labels for ImageFormat · 20
  LabelType element · 18
  mail class for · 22
  mail, about insurance · 27
  mail, about Rubber Stamp fields · 138
  mail, automated pickup · 136
  mail, contents type · 30
  mail, sender name · 33
  mail, value required for · 28, 58
  options for shipping · 141
  reply postage · 24
  required, country of origin · 29
  Return Receipt not supported · 27
  shipments, using &customsid · 11

L

Label Server Agreement, receipt and completion of · 9
LabelRequest
  XML elements, table of · 18
  XML syntax · 17
  XML tags for · 17
  XML, example of · 41
LabelRequestResponse
  XML elements, table of · 42
  XML syntax · 42
  XML tags for · 42
  XML, example of · 45
LabelRequestXML.pl, source example · 178
labels
  about printing · 7
  about printing shipping · 7
  examples of, stealth · 139
  supported for tracking · 7
LabelServiceClient.vb, source example · 207
LabelServiceInvoke.cfm, source example · 171
locality and country listing · 215

M

mailpiece shape chart · 143
misprints, requirements for refunds · 140
MultiLocationCarrierPickup
  about · 127
  API · 109
  XML elements · 127
  XML, syntax example · 127
MultiLocationCarrierPickupResponse
  XML elements, table of · 128
  XML syntax · 127
  XML, example of · 129

N

notifications, customize e-mail · 141
nuSoapLabelRequest.php, source example · 182
nuSoapMLCarrierPickup.php, source example · 201
nuSoapRefund.php, source example · 197
nuSoapStatus.php, source example · 198
nuSoapUserSignup.php, source example · 202

P

packet size, enforced limits on · 137
partner path, develop and deploy Endicia Label Server · 9
Pass Phrases
  about · 219
  change · 51
  production · 9

requirement to change · 9
reset · 140
resetting accounts · 9
steps to take · 8
temporary · 9
temporary, change · 16
payments, paid through accounts · 7
Perl examples, source · 178
PHP source examples · 180
PHP5CalculatePostageRate.php, source example · 193
PHP5LabelRequest.php, source example · 184
PHP5LabelRequestInternationalCustoms.php source example · 189
php5RefundRequest.php, source example · 204
pickup
automated carrier · 136
confirmation number · 118, 128
confirmation number for · 119
confirmation number to cancel or change · 109, 114, 124
confirmation number to change · 123
missed, confirrmation number of · 129
PICNumber
request a refund based on · 196
used with status requests · 91
PieceID, used with status requests · 91
placeholders
HTTP POST request and response · 16
HTTP POST request for BuyPostageXML · 46
HTTP POST request for CalculatePostageRateXML · 54, 64
HTTP POST request for ChangePassPhraseXML · 51
HTTP POST request for GetAccountStatusXML · 72
HTTP POST request for GetPostageLabelXML · 17
PLANET code, First-Class tracking · 7
policy, refunds · 10
PostageRateRequest
about · 54, 64
XML elements, table of · 54, 64
XML syntax · 54, 64
XML tags for · 54, 64
XML, example of · 60, 67
PostageRateResponse
XML elements, table of · 60, 68
XML syntax · 60, 68
XML tags for · 60, 68
XML, example of · 63, 69
production
moving to · 9
qualifying for · 9
startup steps · 10
production account, sign up for · 16
production Pass Phrase · 9
production server · 9, 219
proxy server · 137

Q

qualifying for production · 9
questions and answers · 140
quick start guide · 6

R

reactivate account · 140
RecreditRequest
about · 46
XML elements, table of · 46
XML syntax · 46
XML tags for · 46
XML, example of · 47
RecreditRequestResponse
XML elements, table of · 48
XML syntax · 48
XML tags for · 48
XML, example of · 49
RefundRequest
about · 77
error codes · 80
XML · 78
XML elements,table of · 77
XML syntax · 77
XML tags for · 77
RefundResponse
XML elements, table of · 78
XML syntax · 78
XML tags · 78
XML, example of · 79
refunds
based On PICNumber, source example · 196
policy · 10
requirements for · 140
Request a First-Class Shipping Label
source example · 171
source example · 187
Request a Label Refund, source example · 204
Request a List of Transactions, source example · 175, 199
Request a New User Signup, source example · 202
Request a Refund
Based On PICNumber source example · 196
C# source example · 173
source example · 195, 197
Request a Shipping Label
example of source · 207
source example · 205
source example · 180, 184
Request a Shipping Rate, source example · 193
Request an International Label, source example · 189
Request the Status of a Shipment, source example · 198
Requester ID · 219
Requesting a Carrier Pickup API · 109
Requesting a Refund API · 77

Requesting a Shipping Label, API · 17
Requesting Package Status API · 91
Requesting Transaction Lists API · 97
requirements, system · 6
Rubber Stamps
  about · 138
  elements · 31
  international mail, using · 138
Ruby source example · 205
RubyCURLLabelRequest.ruby, source example · 205

S

sample label
  send e-mail message · 140
  where to send by email · 9
SCANRequest
  about · 104
  error codes · 108
  Web Service · 104
  XML elements, table of · 104
  XML tags for · 104
  XML, example · 106
SCANRequest XML syntax · 104
SCANResponse
  XML elements, table of · 106
  XML syntax · 106
  XML tags for · 106
  XML, example of · 107
scans
  bulk acceptance · 136
  using forms · 104
Scheduling a Carrier Pickup API · 111
Scheduling a Multiple Location Carrier Pickup Request API · 127
server
  production · 9
  proxy · 137
sign up
  accounts · 11
signature
  confirmation · 28, 57, 67
  confirmation fee · 44
  confirmation, fee · 62, 69
signature confirmation · *See also* confirmation.
  eligible matter for · 137
Signing Up for an Account API · 82
SOAP
  example of shipping label request · 182
  example, request carrier pickup · 201
  example, request international label · 189
  example, request label refund · 204
  example, request new user signup · 202
  example, request refund · 196
  example, request refund using Refund Web Service · 197
  example, request shipment status · 198
  example, request shipping label · 184
  example, request shipping price · 193
  format, send label requests using · 6
  invalid format of request · 169
  limits on packet size · 137
  protocol, send requests using · 6
  submission, example using · 80, 90, 96, 108, 116, 120, 125, 130
  what is · 16
solutions, common · 140
source · *See* examples.
  examples · 170
source examples
  caution statements · 170
  Coldfusion · 171
SSL/HTTPS, accessing Endicia Label Server Web Service through · 137
stamps
  printable postage · 136
Stamps
  Rubber · 138
startup steps, production · 10
state code
  invalid, Dial-A-ZIP · 167
  of missed pickup · 129
  of pickup · 128
status
  accounts with Active · 72
  of accounts request · 72
status flag
  account · 48
  account status, active · 75
status requests · *See also* StatusRequest.
  PICNumber or PieceID · 91
  submit Web Service · 91
  submit, more efficiently · 91
  succeeded or failed · 75
  to uniquely identify account · 74
StatusCode Definitions · 96
StatusRequest · *See also* status requests.
  about · 91
  error codes · 96
  XML · 92
  XML elements, table of · 92, 94
  XML syntax · 91
  XML, tags for · 91
StatusResponse
  XML syntax · 94
  XML, tags for · 94
StatusResponse XML, example of · 95
stealth
  postage element · 24
  postage,example of label · 139
Sunday service, Express Mail · 137
support, contacting · 12
system requirements · 6

T

tables
- AccountStatusRequest XML elements · 74
- AccountStatusResponse XML elements · 74
- Carrier Pickup Request API · 111
- CarrierPickupCancel XML elements · 117
- CarrierPickupChangeResponse XML elements · 123
- CarrierPickupResponse XML elements · 113
- ChangePassPhraseRequest XML elements · 51
- GetTransactionsListing Response XML elements · 100
- GetTransactionsListing XML elements · 98
- LabelRequest XML elements · 18
- LabelRequestResponse XML elements · 42
- MultiLocationCarrierPickupResponse XML elements · 128
- PostageRateRequest XML elements · 54, 64
- PostageRateResponse XML elements · 60, 68
- RecreditRequest XML elements · 46
- RecreditRequestResponse XML elements · 48
- RefundRequest XML elements · 77
- RefundResponse XML elements · 78
- SCANRequest XML elements · 104
- SCANResponse XML elements · 106
- StatusRequest XML elements · 92, 94
- UserSignupRequest XML elements · 82
- UserSignupResponse XML elements · 86

Test Server · 219
tracked mail, supported mail classes for · 7
TransactionsListingResponse
- XML syntax · 100
- XML tags · 100

U

unknown return code, Dial-A-ZIP · 167
UserSignupRequest
- about · 82
- XML · 85
- XML elements, table of · 82
- XML syntax · 82
- XML tags for · 82

UserSignupResponse
- XML elements, table of · 86
- XML syntax · 86
- XML tags for · 86
- XML, example of · 88

Using the SCAN Form API · 104
USPS
- error codes · 116, 119, 120, 124, 126, 131
- requirement to change Pass Phrase · 9

V

validate
- address, unable to · 165
- addresses · 132
- all addresses · 25

VisualBasic.NET examples · 207

W

Web service
- invoke XML · 6
- XML · 6

Web Service
- about · 6, 7
- about implementing · 8
- automated signup · 82
- automated submission refund requests · 77
- Carrier Pickup Request · 201
- Description Language (WSDL) · 6
- Endicia Label Server · 15
- for account sign-up · 11
- get transaction listings · 97
- interface · 7
- invoke XML · 6
- Refund · 77
- Refund, API · 195, 196, 197
- Refund, example of · 173
- SCAN Request service · 104
- sign up for accounts · 82
- SOAP-based · 16
- Status, API · 198
- submit pickup requests · 109
- submit status requests · 91
- use default values · 77
- used, to sign up account · 11

WSDL, about · 6

X

XML
- requirement to parse · 6
- Web Service · 6
- Web Service, invoke · 6

XML tags
- CarrierPickupChangeResponse XML · 124
- CarrierPickupResponse · 113
- for AccountStatusRequest · 73
- for CarrierPickupCancelResponse · 118
- for CarrierPickupChangeResponse · 123
- for CarrierPickupResponse · 127
- for ChangePassPhraseRequest · 51
- for ChangePassPhraseRequestResponse · 53
- for LabelRequest · 17
- for LabelRequestResponse · 42
- for PostageRateRequest · 54, 64
- for PostageRateResponse · 60, 68
- for RecreditRequest · 46
- for RecreditRequestResponse · 48

for SCANRequest · 104
for SCANResponse · 106
for StatusRequest · 91
for UserSignupRequest · 82
for UserSignupResponse · 86
GetTransactionsListing · 97
RefundResponse · 78
StatusResponse · 94
TransactionsListingResponse · 100

Z

ZIP Code
invalid, Dial-A-ZIP · 167
of pickup · 128