



Stamps.com Web Services

Developer Guide v1.0

13 November 2012

Stamps.com Confidential

Confidential, unpublished property Stamps.com, Inc. Use and distribution is limited solely to authorized personnel.

This document may contain confidential and privileged material for the sole use of the intended reader. Any review or distribution by others is strictly prohibited. If you are not the intended reader please contact the author and delete all copies.

The use, disclosure, reproduction, modification, transfer, or transmittal of this work for any purpose in any form or by any means without the explicit written permission of Stamps.com, Inc. is strictly prohibited.

Contents

1	An Introduction to SWS/IM	3
2	Getting Started.....	4
3	Some Concepts Unique to USPS and Stamps.com.....	5
4	The Anatomy of an SWS/IM Integration.....	6
5	Authentication	8
6	Purchasing Postage	10
7	Preparing to Ship.....	12
8	Generating a Shipping Label	15
9	Post-Print Features – The User’s Print History	16
10	GetURL	18
11	Error Handling.....	20
12	International and APO/FPO Shipping.....	22
13	Important Best Practices.....	23
14	Appendix A – Services and Special Services Supported	25
15	Appendix B – Common SWS/IM Web Methods	26
16	Appendix C – Integration Completion Checklist	28

1 An Introduction to SWS/IM

SWS/IM or **Stamps.com Web Services for Individual Meters** is our SOAP based API for developers who wish to add USPS shipping to their applications. Using SWS/IM your application can get USPS shipping costs, verify addresses, generate a shipping label and check the tracking status of packages that you or your users have shipped. This document is a primer intended to help you understand the organization and capabilities of SWS/IM and how to begin developing your application¹.

What USPS Services are supported by SWS/IM?

SWS/IM supports all major domestic and international USPS services including First Class Mail, Priority Mail, Express Mail, Media Mail and Parcel Post. For a full list of services, special services and flat rate options supported refer 'Appendix A – Services and Special Services Supported'.

What types of labels can I print using SWS/IM?

SWS/IM is primarily meant to support package shipping. You can generally print domestic shipping labels (4x6) and international shipping labels (with customs forms). Most labels usually come with tracking².

Currently we do not offer developers the ability to print NetStamps labels (the bar-coded version of USPS stamps) or postage printed directly on envelopes. Letter or Postcard mail is therefore not currently supported. Flats (or Large Envelopes) and Packages can be shipped using shipping labels.

What development platforms does SWS/IM support?

SWS/IM is fully platform independent. Your application can be developed on most major IDEs, programming languages or platforms. Your application can be written in any of the Microsoft technologies including C#.net and VB.net or can be a web application built using PHP, Perl or Ruby. Most development platforms provide some form of toolkit for easy SOAP development.

What web methods are supported by SWS/IM?

For a list of the most commonly used web methods refer 'Appendix B – Common SWS/IM Web Methods'. For a full list of web services and object types please refer the Reference document provided to you by your Stamps.com contact or the Stamps.com tech support team.

Can Stamps.com provide code samples of applications?

Given that developers could be working on a wide variety of platforms we do not provide comprehensive code samples for all programming languages and platforms. However, we may be able to provide limited code samples for C#.net, VB.net or PHP.

¹ This document describes a development path for an application with a user interface. However, the same principles can be applied to implement applications without a user interface with application logic and configuration settings supplanting user decisions.

² Tracking is not available for First Class Domestic Flats or all First Class International mail and is optional for First Class Domestic Packages.

2 Getting Started

To begin developing your SWS/IM integration you will need the following –

1. **The Stamps.com Web Services Reference Document** – This will provide detailed information on the various services offered in a human readable format that aids the developer. Please contact tech support at sws-support@stamps.com if you require the latest version of this document.
2. **The SWS/IM WSDL (Web Services Description Language)** – The WSDL³ can be imported or referenced on most popular development platforms to simplify development. Instead of constructing the XML required to make SOAP requests and responses you can treat web services and their associated input and output parameters as standard methods and objects in your development environment. E.g. if you are developing using Visual Studio, if you add a web reference in your project that points to the WSDL you will notice that it will have the same effect as importing a class with its objects, methods and other datatypes. Please contact tech support at sws-support@stamps.com if you require the latest version of the WSDL.
3. **A Stamps.com Integration ID** – This is a unique identifier that represents your application. Your application makes will need to pass this integration ID when authenticating a user.

To get an Integration ID please contact your Stamps.com contact or the Stamps.com customer support team at sws-support@stamps.com. Once we have received and reviewed your request we will provide you with your own Integration ID and work with you to set up your access to our test environment.

4. **Access to Our Test Environment** – Our test environment is a complete replica of our Production (live) environment and will enable you to develop your application without incurring any costs in terms of paying for the postage or subscription fees to Stamps.com.

Once you have completed your integration and meet certain basic requirements⁴ your Integration ID will be approved for migration to our Production environment. At this point you will be provided a production WSDL. The Production WSDL is identical to the test WSDL except for that it points to our Production server.

5. **Stamps.com Username and Password** – Every user who prints postage using Stamps.com must have a Stamps.com account. All of your users will therefore need to be registered Stamps.com customers or will need to register for a Stamps.com account before they can start using the features in your software that are integrated into SWS/IM.

NOTE: To aid development we offer developers a Stamps.com account on our Test environment. Postage printed using this account will be very similar to live postage but cannot be used. Please make sure to either not print postage generated using the test account or carefully destroy any prints generated.

³ Refer <http://www.w3.org/TR/wsdl> or http://en.wikipedia.org/wiki/Web_Services_Description_Language for more information on WSDLs.

⁴ For more details on the requirements for migration to production refer Appendix C – Integration Completion Checklist.

3 Some Concepts Unique to USPS and Stamps.com

As you may have already read in the prior section, to use your integrated application, your users will need to either already have a Stamps.com subscription account or sign-up for one. This is not dissimilar to a FedEx or UPS account that businesses are usually required to hold when shipping with them. However, there is one important difference between using PC Postage⁵ for the USPS and shipping with other carriers. It probably helps to get this out of the way before we proceed further.

USPS Postage is Prepaid

Unlike with FedEx or UPS when you print a USPS shipping label your customer has already paid for it. What does this mean for you?

- **Buying postage and printing postage are separate actions** – E.g., when our customers use our desktop client software they first stock up on postage and create a balance. They then print postage and draw down from this balance. This is a security measure required of us by the USPS.

We therefore offer two separate web services – to purchase postage and to generate a label. It helps to provide a “Buy Postage” feature that enables your users to increase their postage balance when it gets low.

- **Offer the ability to Reprint** – It usually helps to provide some capability to print a shipping label again in case the customer was not able to print it properly the first time. The customer cannot simply void the label and print another one.
- **Offer the ability to Refund** – Although the customer cannot “void” a USPS label printed using Stamps.com the customer can request a refund for the label. SWS/IM provides the ability to request refunds through a web method. Refunds are not approved immediately and usually take up to 2 weeks when tracking is included with the label. The USPS requires this delay so they can check to make sure that the label was not actually used by the customer. If tracking is not included then refunds can take up to 30 days and the customer is usually required to mail the label in as evidence that this was not used.

Stamps.com takes great pains to make this process as easy as possible for the user as possible and we make it easy for you to offer these capabilities to your/our customers.

⁵ PC Postage is a (now archaic) reference to postage printed on PCs. This is common terminology to reference the technology offered by Stamps.com.

4 The Anatomy of an SWS/IM Integration

All integrations in to SWS/IM have to implement two basic groups of functionality –

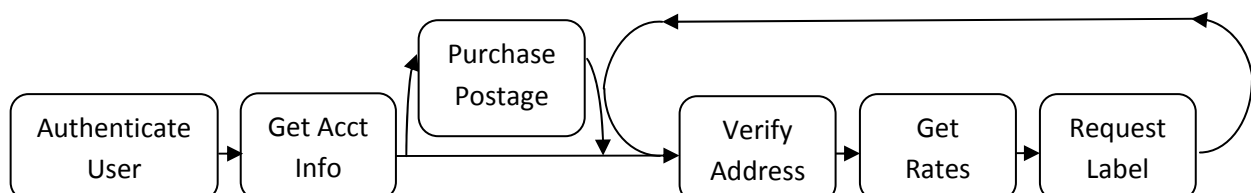
- **Printing Postage** – This is the primary purpose of almost all SWS/IM integrations
- **Post-print Functionality** – Once the user has printed postage they will need to perform a variety of actions on their prints including requesting refunds and tracking packages.

Printing USPS Postage –The Basic Necessities

Any SWS/IM integration must at a minimum implement the following features –

1. **Authenticate a User** – This involves accepting the user’s username and password and verifying it with Stamps.com servers. This is usually done once at the beginning when the application is launched. You can either use a login screen that is shown at the time of launch or you can save the settings and log in the user in when they launch the application. The latter usually offers lower security, so the former is the recommended behavior.
2. **Get User’s Account Information** – This provides a lot of information about the status of a Stamps.com user account including postage balance and user access restrictions. It can help you anticipate a user’s needs and capabilities and build a great user experience. However, when using it, you should exercise caution as it reaches out to a number of our servers and can really slow your application down.
3. **Purchase Postage** – As described in the prior section, users will have to purchase and maintain a postage balance. Most Stamps.com subscriptions usually come with a small amount of free postage for the user (usually \$5) but to continue printing postage the user will have to purchase postage from time to time.
4. **Generate a Shipping Label** – To print postage you have to go through a 3-step process.
 - a. **Verify Address** – This step helps ensure that a valid and properly formatted recipient address is being used. This step allows the USPS to process packages quickly and easily.
 - b. **Get Rates** – In this step you request the various USPS services available to the user. In SWS/IM this will require a minimum of an origin and destination zip codes. You then show the user the options they have to ship with the USPS, the cost involved and the time it can take to ship a package from origin to destination.
 - c. **Request a Label** – Once the user has selected a USPS service you provide the user’s selection to an SWS/IM web method along with the verified address to receive a shipping label.

The following diagram shows the sequence of the above operations –



At this point the user's primary requirements have been met. However, most, if not all, applications will have to implement one or more additional functions to provide the user with a full range of postage capabilities.

Post-Print Functionality – High Value Features

In addition to postage generation, most applications should also implement a set of features that allow a user to act on postage they have already printed –

1. **Maintain a Print History** – The user should be able to view a list of all their postage prints so they can perform a variety of actions including request refunds or tracking packages. You should keep a log of all postage printed by the user within your application so that the user can review their print history.
2. **Request Refunds** – As explained previously the user should have the ability to request a refund within your application.
3. **Track Packages** – Users will need to track packages and this can be done through a SWS/IM web method as well.
4. **Other Post-Print Features** – Users may also require other capabilities such as generating SCAN Forms⁶ or filing insurance claims (for insured packages).

Stamps.com offers two paths to implementing the above post-print features.

- **Build Your Own User Interfaces for All Aspects of the Integration**

One option is for you to develop all the user interfaces necessary to provide all of the features described above. Stamps.com has web services you will need for a full featured application. You can therefore completely control the user experience and ensure that every user interface looks just the way you want it. This gives you maximum flexibility but requires a little more development effort.

To build an application using this method read Sections 5 through 9 (you can skip Section 10).

- **Build the Postage Printing User Interfaces and Use Our Web-Based Interfaces for the Rest**

A quicker path to completion would be to use our existing user interfaces for most of the post-print functionality. We provide out-of-the-box web-based user interfaces that you can provide to your users. These interfaces require very little development on your end. However, these interfaces cannot be modified in any way, even cosmetically. You will still need to implement user interfaces for purchasing and printing postage, but pretty much everything else you can simply redirect the user to our web pages.

To build an application using this method read Sections 5 through 8 and Section 10 (you can skip Section 9 although there is information you may want to consider there).

⁶ SCAN form or Shipment Confirmation Acceptance Notice is described in detail in the section *Post-Print Features – The User's Print History*.

5 Authentication

The first step your application will have to take is to log a user in. To do so you will use the `AuthenticateUser` web method.

Input Parameters

- Your *Integration ID*
- A valid Stamps.com *Username* and
- The *Password* for the Username

Output Parameters

- Authenticator token

If an Authentication is successful the request will return no error. In addition it will also provide an Authenticator token. This token allows you to begin making requests to the various SWS/IM web methods.

The Authenticator Token

The Authenticator token is evidence that the user has been authenticated. It is a temporary passcode that is passed back and forth every time you call a SWS/IM web method. Most SWS/IM web methods accept an Authenticator token as an input parameter and return a new (replacement) Authenticator token as one of the output parameters⁷. An Authenticator token is therefore only valid for the next web method call. Your application should save the Authenticator token provided in the response to the last web method call it made and use it for the next call. This method provides two benefits –

1. It improves the security of the user's credentials as the username and password are not passed back and forth each time.
2. Since authenticating a user on our server is a time-intensive process it also speeds up the response times to your requests when you use the token.

An Authenticator token is valid for the session of communication between your application and the SWS/IM server. If your application is idle for an extended period of time then the Authenticator token will expire. At this point, you will get a "conversation-out-of-sync" error when you make an SWS/IM call informing you that the token has expired. You should then log the user back in using `AuthenticateUser` and use the fresh token to continue⁸.

It is important to note that all SWS/IM methods also accept the user name and password in lieu of the Authenticator token. However, it is recommended that you use the Authenticator token instead as using

⁷ The exceptions are `StartPasswordReset`, `FinishPasswordReset` and `AuthenticateUser`.

⁸ Authenticator tokens can also expire if a username is used by another instance of your application or another application that is integrated in to SWS/IM.

the username and password each time will slow down the time taken to respond to your request as we attempt to authenticate the user each time.

6 Purchasing Postage

As mentioned earlier users will have to buy postage and draw down a balance before printing. There is no minimum amount that users are required to maintain so users generally do not have to buy postage in large amounts. However, Purchasing postage is an asynchronous task (you will have to make the request and then check through a second web method to see if the request was successful). This means that purchasing postage in small quantities or purchasing just before a label is about to be printed can slow down the user's experience. A good solution is to offer the user various purchase amounts that they can select from (\$10, \$20, \$30, \$40, \$50 or other). To purchase postage you will need to implement the following two web methods –

PurchasePostage

You should use the PurchasePostage web method to make a request to purchase postage.

Input Parameters

- Authenticator
- PurchaseAmount
- ControlTotal

Output Parameters

- Authenticator
- PurchaseStatus
- TransactionID
- PostageBalance
- RejectionReason

Apart from the amount of postage the user wishes to purchase you should send the ControlTotal. This is the amount of postage that the user has consumed over the lifetime of the account. Sending SWS/IM this ControlTotal lets the server know if you have the correct postage balance when buying postage. If you do not then it is possible that the postage balance was changed through other means (the user could have logged in through another integrated application to purchase or print postage). Initially you can get this ControlTotal by calling GetAccountInfo (see 'Important Best Practices' for more information on the usage of GetAccountInfo). Thereafter, you can track the Control Total by adding the postage consumed with each print to the previous ControlTotal.

The first thing to do when you receive a response to this method is to check the PurchaseStatus. This can be Success, Pending, Processing or Rejected.

- **Success:** If the status is Success then you can inform the user of the successful purchase. The PostageBalance field is the new balance in the user's account.

- **Pending, Processing:** If the status is either Pending or Processing then you'll have to call GetPurchaseStatus to see if the purchase went through successfully. Use the TransactionID returned by this call when calling GetPurchaseStatus.
- **Rejected:** If the status is Rejected then check the RejectionReason to see why the purchase was rejected. If it was because of a stale (incorrect) ControlTotal then show the user the new PostageBalance and check to see if they would like to continue purchasing postage. If it was rejected for any other reason you should inform the user.

GetPurchaseStatus

For a variety of reasons a purchase may not go through instantaneously including the amount of time it may take for us to reach out to a payment gateway to authorize a credit card purchase. If a postage purchase is pending you should use the GetPurchaseStatus web method to check the status of the PostagePurchase request.

Input Parameters

- Authenticator
- Transaction ID

Output Parameters

- Authenticator
- PurchaseStatus
- PostageBalance
- RejectionReason

When calling this method you should provide the Transaction ID returned by the PurchasePostage call. In return you will get a response similar to the PurchasePostage call. You should handle the output of a GetPurchaseStatus call the same way you would a PurchasePostage call (check the PurchaseStatus and respond appropriately). One thing to keep in mind when calling GetPurchaseStatus is that you may get a pending or processing status (more likely a processing status) again. If this happens then you will need to call GetPurchaseStatus again. Use an exponential backoff algorithm as you loop through the GetPurchaseStatus call (increase the wait time between each time you call GetPurchaseStatus). Under most circumstances you should get a Success or Reject status within a second but you should probably not timeout on the transaction for at least 15 seconds before you give up.

7 Preparing to Ship

Once a user has been authenticated, your application is probably designed to identify what is being shipped and to which destination. At this stage there are two SWS/IM web methods you are going to need –

- CleanseAddress – This will verify the destination address.
- GetRates – This will provide you with a list of services available to the destination.

CleanseAddress

CleanseAddress as the name suggests provides address verification functionality.

Input Parameters

- Authenticator
- Address

Output Parameters

- Authenticator
- Address (a cleansed version if the call was successful)
- AddressMatch (a Boolean flag)
- CityStateZipOK (a Boolean flag)

SWS/IM requires you to verify the address for all domestic shipments⁹. Once you have the destination address call CleanseAddress, this will verify the address is valid and also make formatting changes to meet USPS addressing standard.

When you receive the response to your request the first thing to do is check the AddressMatch field that is returned. This is a Boolean value that is True if an exact match was found. If AddressMatch is false, check to see if the CityStateZipOK field is true. If this is true that means the street address provided could not be verified but the city, state and the zip match each other. In this case, you can continue to print postage for this address but it is probably worth informing the user that the street address could not be verified. If both fields are false then you should flag an error as SWS/IM will not allow you to proceed. Your algorithm for address verification should therefore look something like this –

If (**AddressMatch** is False)

 If (**CityStateZipOK** is True)

 Warn user that the street address could not be verified

 If the **warning** is ignored

⁹ Addresses cannot be verified for international shipments but we do verify the destination country. Please ensure that you use a country name from the accepted list published here - <http://pe.usps.com/text/imm/immctry.htm>.

```
        Proceed to the next step in printing postage
    Else
        Repeat address verification with any changes the user has made
    End If
Else
    Show an error to the user that the address could not be verified
End If
Else
    Proceed to next step in printing postage
End If
```

NOTE: *CleanseAddress does not necessarily have to be called before proceeding to GetRates, but it should be completed before you proceed to request a Shipping Label.*

GetRates

In general, your workflow should be to call GetRates to get a list of Services available to the user, let the user make a choice (or make one for the user based on preset rules) and request a shipping label for the Service selected. To show the users what shipping options are available to them you will need to call GetRates.

Input Parameters

- Authenticator
- Rate

Output Parameters

- Authenticator
- Rates (this is an array of Rate objects)

The key to understanding the process here hinges on understanding the Rate object. The SWS/IM Rate object represents a USPS Service along with all the bells and whistles such as Signature Confirmation that you can add on to it. The Rate object contains the Service Type, Package Type, weight information, USPS's service cost and service level estimate (in days) amongst other things. GetRates accepts a Rate object as input and provides an array of one or more Rate objects that match your input Rate object as output. You can fill one or more fields in the Rate object you provide to GetRates and in response it will return all the USPS services that meet your criteria. E.g., if you want all Priority Mail options then you should set the ServiceType field to Priority Mail. In response you will get all mailing options that are Priority Mail including Flat Rate options such as the Flat Rate boxes.

When calling GetRates keep in mind that if you specify very few values in the input Rate object you will get a large number of Rate objects in the output Rates object. It is best to provide as much as info as possible when calling GetRates. E.g., if you know that the user wants to ship a Small Flat Rate box then set this in the PackageType field and Priority Mail in the ServiceType field, this will limit the response to exactly what your user is looking for. The simplest way to filter the Rates you receive would be to set the From and To Zip for domestic packages and the destination country for international packages.

NOTE: The only required field for GetRates is ShipDate in the Rate object, but consider passing the From and To Zips at a minimum.

AddOns

AddOns are Special Services such as Signature Confirmation, Hidden Postage, Insurance, Registered Mail and etc. AddOns available can vary based on the Service type that the user chooses. Dealing with AddOns can be tricky as the USPS requires that certain AddOns be included when shipping with certain Service Types. Other AddOns can be optional but may carry an additional cost that you will need to inform the user. SWS/IM simplifies all of this for you.

When you receive a response to GetRates, each Rate object in the Rates array comes with its own AddOns¹⁰. Each Rate object in the response also comes with its own list of required AddOns, you can see this list in the RequireAllOf array in the Rate object. So here's how you would go about it –

- **Step 1** – Make a call to GetRates. Make sure to be as specific as possible when making the GetRates request.
- **Step 2** – Cycle through all the Rate objects returned and identify the one you wish to use. You can do this either by showing the user a list of rates returned or using preset rules to identify the Rate that you want.
- **Step 3** – Look up the RequireAllOf array. All the AddOns specified here have to be included when requesting a shipping label.
- **Step 4** – Look up the AddOns array. This has a list of all AddOns associated with the Rate object. Any AddOns in this list that are not in the RequireAllOf array are optional.

At this point you should have identified the specific Rate object you want to generate a label for and the AddOns that you wish to include along with it. This Rate object that you construct will be the primary input parameter for the next web service you will need – CreateIndicum.

¹⁰ AddOns is an array within the Rate object.

8 Generating a Shipping Label

To generate a shipping label you should use CreateIndicium.

Minimum Required Input Parameters

- Authenticator
- Rate
- From (Address)
- To (Address)

Output Parameters

- Authenticator
- Rate (with cost)
- TrackingNumber
- URL

CreateIndicium generates all the labels you will need, whether you are looking to ship domestic or international. This section gives you a basic primer on how the method can be used to generate a basic domestic shipping label. For more information on how to process international labels with customs forms refer Section 12 – International and APO/FPO Shipping. To create a domestic shipping label you will need to provide the Rate object you created (see GetRates in Section 7 – Preparing to Ship). You will also need to provide the From Address and the To Address. In response you will get a Tracking Number (if tracking is included) and a URL. The URL links to an image of the shipping label that you requested.

You may notice that SWS/IM supports a CreateTestIndicium. This is actually an old web method that will be deprecated soon, you should therefore not use it to generate sample labels. To create a sample label you can call CreateIndicium and set the SampleOnly input flag to True. This will generate a sample that is similar to the live postage you are requesting but without any valid postage. This can be useful for testing your application on production as well as for providing the user the ability to print a sample label.

Note: Use the ImageType input parameter to specify the file format that you would like the shipping label in. You can also convert the file format after making the request. E.g. if you get a PNG image of the file, you can simply change the file extension in the URL you receive to .pdf to get the PDF version of the image.

9 Post-Print Features – The User’s Print History

Once you’ve generated a shipping label you’ve pretty much completed the basic SWS/IM integration. There are a number of features you will still need to provide but the process gets easier from here on. The most important features you will have to implement are related to the user’s print history. These include –

Refunds

Users can request a refund for shipping labels that they are not going to use. Refunds are generally requested electronically, the user does not normally have to send any evidence to collect a refund. The USPS will monitor tracking scans to see if the label the user is looking to refund has been used. If they do not find tracking scans for a couple of weeks the postage will get added back to the user’s account automatically. Refunds can be requested up to 10 days after a label has been printed. To request electronic refunds you can use the CancelIndicum web method. You will need to pass either the Transaction ID or the Tracking Number. As noted previously, these two values are provided in response to a CreateIndicum call.

NOTE: *The exception to the electronic refund process is for shipments that do not include tracking numbers. In these cases the user has to request a mail-in refund which currently can only be done from their Stamps.com Print History on our website. However, we allow you to embed this page in your application (see Section 10 – GetURL below for more details).*

Carrier Pickup

The user may want to have their packages picked up at their office (or shipping location). You can request a USPS pickup. To do so the user must have shipped at least one domestic Priority or Express Mail package or any international package. To request a carrier pick up you should use the CarrierPickup web method. You will need to provide the name and address where the packages are to be picked up and the number of Priority, Express and international packages to be picked up. If the request is successful you will receive a date and time of pickup along with a confirmation number.

Tracking

Users will want to know where their package is en route to the destination and what the status is. You can get a list of tracking events for any shipment by calling the TrackShipment web method. You will need to provide either the Transaction ID or the Tracking Number on the shipping label. Both of these values are provided to you in response to a CreateIndicum call¹¹.

¹¹ You will not receive a tracking number when shipping First Class Flats or turning off tracking for First Class, Media Mail or Parcel Post packages. You cannot track these packages using TrackShipment.

SCAN Forms

SCAN Forms (Shipment Confirmation Acceptance Notice forms) are a useful tool for users who mail a number of packages together. In many instances, if a user delivers a number of packages to USPS at once, the packages are not scanned immediately. This means that the recipients of the packages cannot see that their shipments are already in the mail. In some cases the package may never be scanned until the USPS actually delivers them.

To ensure that a package appears in the mail stream and shows tracking scans the USPS came up with the concept of SCAN Forms. A SCAN Form has a list of packages with tracking numbers and a single barcode that represents all of the tracking numbers. When a USPS representative scans this barcode, all the packages in the SCAN form immediately show up as having been received by the USPS.

You can generate a SCAN Form using the CreateScanForm web method. You will need to provide a list of Transaction IDs along with the From Address of the packages. In response you will get a SCAN Form ID and a URL. You can download an image of the SCAN Form from the URL and print it out for your user.

NOTE: *You can request the SCAN Form in a variety of image formats. You can also specify if you want to print an instruction sheet for the customer.*

10 GetURL

As mentioned in the previous section there are a number of features a user will require once postage is printed. In addition, there are several account management features that a user will need. SWS/IM simplifies all of this for you. The GetURL method will return a URL to a variety of features for which we have a web-based user interface. You can simply redirect the user to the URL returned by GetURL and the user won't be required to log in.

Input Parameters

- Authenticator
- URLType

Output Parameters

- Authenticator
- URL

The URLType parameter specifies the web page that you are requesting. Some of the commonly needed URLs are¹² -

URLType	Description
OnlineReportingHistory	Users can search the history of all their prints from this page. You can show the user this page instead of implementing the TrackShipment web method.
OnlineReportingRefund	This web page allows users to search for transactions that can be refunded and request a refund. You will need this web page for labels printed without tracking. These labels are not electronic refunds and have to be mailed. The user will be required to print a refund request form and send it in and all of this is made available from within this page.
OnlineReportingPickup	This web page allows the user to schedule a pickup with the USPS. You can show the user this page instead of implementing the CarrierPickup web method.
OnlineReportingSCAN	This web page allows the user to print a SCAN form. You can show the user this page instead of implementing the CreateScanForm web method.
OnlineReportingClaim	This page allows the user to file a claim for any packages that have been insured.
StoreChangePlan	This page allows the user to upgrade or downgrade their account. Upgrading their Stamps.com account gives them additional postage features.
ReportsBalances	This page shows the use a report of their postage balance, how

¹² For a full list of pages that you can access refer the SWS/IM Reference document.

	much they have purchased, how much they have consumed, how much has been refunded and how much they have left. Data is shown by period (day, week, month, year etc.).
ReportsExpenses	This is a report of expenses that the user has incurred. This report shows how much the user has spent on postage, supplies (labels), subscription fees and insurance purchases.
ReportsPrints	This is a convenient report that shows tables and charts that allow the user to view how much postage they've consumed in any period, broken down by the type of postage they've consumed.
StorePaymentMethods	The user can change their credit card or switch to any other payment method allowed for their account through this page.
StoreMyProfile	This is the user's profile page. It is a gateway page to a number of other pages including the password change page. You can implement access to this page instead of implementing individual pages.
StorePassword	As an alternative to the 'My Profile' page, you can build your own version of the page but link to specific pages such as the 'Change Password' page using this URLType.
StoreCommPreferences	This is another page that you can link directly to instead of taking the user to their 'My Profile' page. This page allows the user to turn on or off email communication from Stamps.com.

11 Error Handling

More often than not, from time to time, you are likely to receive errors when calling a web method. In general if a request is successfully made to the SWS/IM server any errors will be returned as a SOAP fault. Errors will always return an error code and usually will return an error message with it as well. Your application should handle SOAP faults and throw an error message based on the error code returned. In general check to see if there is a valid error message returned. If there is, you can show the error to the user along with any additional text you wish to add. If there is no error text and you are not explicitly handling the error code the best thing to do is to fall back to a standard error message with the error code.

Most errors can generally be categorized under 1 of 4 categories.

User input errors

This may happen if the user entered incorrect values. A simple and straightforward example of this is when the user enters alphabets in numeric fields. **Such errors are best pre-empted.** Ideally you should build your application to respect the data types required for each input parameter to SWS/IM web methods. If invalid data types are passed the web method call will not even be made. The request will fail the SOAP schema restrictions and an error will be returned immediately.

User access errors

User access errors occur when a user is attempting to perform an operation that they do not have permission to access. Stamps.com users can be on various subscription plans and depending on their plan they may have access to perform only a restricted set of actions. E.g. A user may be restricted to purchasing only \$250 in postage each time they attempt to purchase postage. If the user attempts to purchase more postage the PurchasePostage call will return an error with a Reject reason. You can handle such scenarios in one of two ways –

- **Pre-emptively:** The GetAccountInfo web method provides details on what the user's access restrictions are (see Section 13 - Important Best Practices for more details on how to use GetAccountInfo). You can store these access restrictions and prevent the user from performing any operations that would violate their restrictions.
- **Reactively:** You can simply make the web method call and let the server return an error. You can then show this error to the user along with a message.

Depending on how frequently you feel the error is likely to occur for your user you can choose either of the two above solutions. Be pre-emptive if you think a scenario is likely to recur a lot and be reactive for edge-cases that are not likely to recur).

Application Errors

These errors occur when the user has entered valid data but your application is not accounting for certain scenarios. The most common such instances are when the application has idled for too long. In these cases the Authenticator may have expired. In such a case you should attempt to rectify the situation by authenticating the user again and re-trying the transaction. If your application does not store the username and password then you should prompt the user to login in again.

Server Errors

Stamps.com provides a very high level of service. Integrators usually get advance notice of scheduled maintenance and unscheduled down times or failures are few and far in between. However, occasionally, you may receive errors from the server that you or the user is not responsible for. In such a case it is likely something that Stamps.com will have to solve on our end. For unexpected errors that occur you should show a standard error message in your software. Depending on how you prefer to handle this you can direct your users to a representative of your company or to Stamps.com customer support. We will be happy to resolve any issues for your users when they are a result of an error on our servers.

12 International and APO/FPO Shipping

By now you should have a good idea of how to integrate into SWS/IM. However, handling international shipping labels or APO/FPO/DPO shipping labels can require a little extra effort.

International Shipping Labels

Customs Forms

The most obvious difference when handling international shipping labels is the customs form. When creating an international shipping label you should pass the Customs object to the CreateIndicium web method. This should contain all the customs information regarding the shipment¹³. In response SWS/IM will return a label with postage and customs form combined.

The 2976 vs the 2976-A

Depending on the USPS service selected there are two different types of customs forms that can be generated – the 2976 or the 2976-A. The 2976 is the short form single page layout. The 2976-A is a full form 3 page layout including instructions. SWS/IM will automatically generate the correct customs form. One thing to note is that by default international labels are returned as PDFs. This is because the 3-page 2976-A can be concatenated into a single 3-page PDF document. However, if you wish to receive the 2976-A label in a different format you can do so and instead of a single URL you will receive three URLs in the CreateIndicium request¹⁴.

APO/FPO/DPO Shipping Labels

APO/FPO/DPO (Army Post Office/Fleet Post Office/Diplomatic Post Office) addresses are international destinations that are treated as domestic addresses. The way these destinations work is that the USPS delivers to specific domestic destinations (in New York, Florida and San Francisco) and the army delivers the mail to the relevant international destination where the recipient is stationed. However, since the mail travels to international military stations APO/FPO/DPO mail requires a customs form.

When generating a label for an APO/FPO/DPO destination SWS/IM generally treats the request as a domestic shipping label request. You will therefore get a 4x6 shipping label in response. However, you have the option to request a customs form in addition to the shipping label. The customs form generated will be separate from the shipping label and unlike the international label will not carry postage. You can request either a 2976 or a 2976-A customs form. To request a customs form for an APO/FPO/DPO shipping label, simply provide the Customs object when making a CreateIndicium request. If you do not generate the customs form using SWS/IM the user will still be required by the post office to manually fill one out when mailing the package.

NOTE: *Freely Associated States, Territories and Possessions of the United States are very similar to APO/FPO/DPO destinations in that they are treated as domestic destinations that require customs forms.*

¹³ See the SWS Reference document for more details on the structure of the customs form.

¹⁴ International labels cannot be requested in EPL or ZPL format.

13 Important Best Practices

There are many ways to go about building an SWS/IM integration. However, there are some practices that are universally good to adhere to.

Use the Authenticator instead of Credentials

Although every Stamps.com web service will allow you to pass the user's user name and password, it is best to use the Authenticator token provided by the previous SWS/IM web service call. This reduces the burden on our servers and speeds up your implementation.

Multiple Shipping Stations on the Same Account

If you are developing an integration to ship your own packages and process large volumes using a number of different stations in the same location you may find that using the Authenticator instead of credentials trips up one shipping station because another station just made a call. In this case the first station no longer has the latest Authenticator and a call to any web service would be rejected. In this case we suggest you build a client-server architecture with a centralized server that makes calls to SWS/IM. Requests from each of the shipping stations can be queued through the server. If you have a handful of stations all making rapid requests during peak hour our tech support or shipping sales team can set you up with multiple user accounts. The server can keep a stack of these accounts and use the first available account to process postage for the next request that comes in from any of the shipping stations. We can link all of your accounts under a single umbrella account so you can still view reports on all of the activity in a single view irrespective of which account you processed the request through.

Use GetAccountInfo Effectively for the Best User Experience

GetAccountInfo is a powerful web service that allows you to set up the context for your application and it is worth the investment to spend some time understanding this web method completely. It provides a lot of information about the status of a Stamps.com user account including postage balance and user access restrictions. GetAccountInfo can help you anticipate a user's needs and capabilities and build a great user experience. However, when using it, you should exercise caution as GetAccountInfo reaches out to a number of our servers and can really slow your application down. GetAccountInfo provides the following information that you will find useful in your application –

- **PostageBalance**

The most commonly needed value returned by GetAccountInfo is the PostageBalance. You may need this value at the beginning when the user logs in and again if the balance you are tracking goes stale (this can happen if another instance of the software logs in using the same account and prints postage).

- **Capabilities**

To build the ideal experience you will need to be aware of a lot of the user's context. E.g. depending on the plan the user has we restrict the amount of postage they can hold in their account. If you

attempt to purchase postage that exceeds this limit then SWS/IM will throw an error. An alternate would be to check what their MaxPostageBalance is and restrict the user from even entering a value that exceeds the limit. While this might be a non-critical case, in other circumstances we may actually restrict certain user accounts from printing shipping labels. Such accounts in all likelihood shouldn't even be used with your software. You can check the CanPrintShipping flag in the Capabilities object. All of this information is available in the AccountInfo object that is a part of GetAccountInfoResponse.

- **LPOCity, LPOState and LPOZip**

There are a number of addresses that can be associated with a user account including the LPO City, State and Zip. LPO refers to Local Post Office and represents the ZIP code from which the user is shipping. This information is available in the AccountInfo object that is a part of GetAccountInfoResponse.

- **Address**

This is the primary contact address for the account and is part of the GetAccountInfoResponse object.

- **CustomerEmail**

This is the primary email address for the account and is part of the GetAccountInfoResponse object.

Consider the following when you use GetAccountInfo:

1. **Use it sparsely:** Use this web method only when you need to. The most obvious occasion to call GetAccountInfo is as soon as the user is authenticated. Another occasion you will need to call it is if the ControlTotal on a PurchasePostage call has gone stale¹⁵. This means that you do not have the correct Control Total and you should call GetAccountInfo which returns this value.
2. **Do not use it repeatedly to check postage balance:** You may want to keep the user updated of their current postage balance. The best way to keep track of the postage balance is to call GetAccountInfo when the user logs in. Thereafter you can keep track of the postage balance through the CreateIndicium call. Every time you print postage CreateIndicium will return the postage balance in the user's account. Only call GetAccountInfo again if PurchasePostage fails because you have the incorrect postage balance.

Never Expose the Integration ID

The Integration ID represents your software and your business. It is a key element in the Stamps.com SWS/IM security model and should never be divulged to anybody except authorized representatives of your business who you are confident will maintain the integrity and security of your relationship with Stamps.com.

¹⁵ Refer Section 6 - Purchasing Postage for more information on the ControlTotal and how it can go stale.

14 Appendix A – Services and Special Services Supported

USPS Services

Domestic

- First-Class Mail
- Priority Mail
- Express Mail
- Parcel Post
- Media Mail

International

- First-Class International
- Priority Mail International
- Express Mail International

Special Services

- Hidden Postage
- Insurance
- Insurance for Registered Mail
- Certified Mail
- Collect on Delivery
- Certificate of Mailing
- USPS Delivery Confirmation
- USPS Express – Sunday / Holiday Guaranteed
- USPS Insurance
- USPS Express – No Delivery on Saturdays
- Restricted Delivery
- Registered Mail
- Return Receipt Requested
- Return Receipt for Merchandise
- USPS Signature Confirmation
- Special Handling
- USPS Express – Waive Delivery Signature
- Do not Deliver on Saturday
- Sunday/Holiday Delivery Guaranteed
- Notice of non-delivery

15 Appendix B – Common SWS/IM Web Methods

While SWS/IM supports a large number of web methods, some are not available for external consumption and a few others are not critical to most applications. The following table describes a list of SWS/IM web methods that you are most likely to need in approximate order of chronological use.

Web Method	Brief Description
AuthenticateUser	This web method is used to log in a user. It returns an Authenticator token that enables you to make the next web method call.
GetAccountInfo	This web method provides you with information regarding the user's account including their postage balance and permissions.
PurchasePostage	This web method allows the user to purchase postage for their account.
GetPurchaseStatus	A postage purchase request may not be authorized immediately. If this happens PurchasePostage returns a "Pending" or "Processing" status. Use the GetPurchaseStatus in these cases to check back in on the status of the PurchasePostage request.
CleanseAddress	Use this web method to verify addresses.
GetRates	Use this web method to get a list of USPS shipping methods available to the user.
CreateIndicium	This web method allows you to request a shipping label for the user.
CancelIndicium	Use this web method to request a refund for postage that has tracking information. This is an electronic refund request and funds will be returned to the user's accounts if there is no evidence that the label has been used. This process usually takes two weeks. To request refunds for labels without tracking user will have to use our web interface to request refunds. This can be done either by logging in to their stamps.com account online or you can provide the same interface using the GetURL web method (see below).
CreateScanForm	This web method allows you to generate a Shipment Confirmation Acceptance Notice (SCAN) Form. The SCAN Form is similar to a manifest allows the user to deliver a number of packages to the USPS and receive acceptance scans for all of them quickly.
CarrierPickup	Use this web method to request the USPS to pick up a set of packages that includes at least 1 Priority Mail, Express Mail or International package.
TrackShipment	Use this web method to receive tracking events for a package that includes tracking.

GetURL	GetURL provides you the ability to embed any of our existing web based user interfaces within your software. GetURL will return the authenticated URL (meaning the user does not have to login additionally) for all of our web interfaces.
---------------	---

16 Appendix C – Integration Completion Checklist

Before migrating your integration from our testing environment to our production environment it helps to take stock of a few things. When you complete your integration please review the following –

The term *Partners* is used for integrators that develop and resell software that uses the Stamps.com API.

1. **Authenticate a User** – You must understand how to call successive web methods using a returned authenticator token.
2. **Purchase Postage** – You must understand that the purchase postage process is asynchronous and should use an exponential back-off wait algorithm. You should use the GetPurchaseStatus web method to ensure a call is successful. For example, the integration should wait 1 seconds before making the first call, then 2 seconds before trying again, and then 4 seconds etc. (more than 4 tries is not necessary).
3. **Generate a Shipping/Mailing Label** – You should email a scanned copy of the **PRINTED** label or should verbally verify the correct sizing of their printed label during the migration meeting.
 - a. You should be able to request (or at least know you can request) the labels in the following formats (all of which are supported by the Stamps.com API).
 1. EPL printer
 2. GIF format
 3. JPEG format
 4. PDF format
 5. PNG format
 6. ZPL format
4. **Verify/Cleanse Address**
5. **Get Rates**
6. **Cancel a Shipping/Mailing Label** – You should understand what labels are e-refundable and what labels require mail-in refunds. This is particularly relevant if you turn off Delivery Confirmation in any scenario.
7. **SCAN Forms** – You should be able to create a SCAN form or have an understanding of how the SCAN form feature works.
8. **Get Account Information** – You should understand that GetAccountInfo should only be called when
 - a. Logging the user in (For *Partners*) (or).
 - b. The postage balance you are tracking has gone stale (or) .
 - c. An operation you thought the user had access to is prohibited by the server.

9. **Hide Integration ID** – Ensure that the Integration ID is never exposed to customers (For *Partners*).