AccuMail® Verify™

Single Record Lookup

Self-Hosted REST Service Package

CONTENTS

PREFACE

The AccuMail Verify Self-hosted REST Service Package provides quick and accurate real-time, single record lookup address correction that can be integrated into your application(s) and websites with ease. You can build applications or websites that accept user entered address information and validate and correct the addresses, based on USPS rules and reference data. This product is an on-premise product (self-hosted) and requires installation on your server. Please see the section labeled Installation Instructions for more information.

WHO SHOULD USE THIS GUIDE?

This guide was developed for companies wanting to implement the AccuMail Verify Self-hosted REST Service into their software applications or web sites and ***assumes a development staff familiar with software programming and web service setup***. It is also intended for network administrators that manage network resources.

TECHNICAL REQUIREMENTS

WINDOWS SERVER

- Microsoft Windows Server operating system (currently supported version)
- 2 Core CPU
- Minimum RAM: 8GB

WEB SERVER CHECKLIST

- Visual C++ 2015 x64 Redistributables installed on the server.
- .NET 4.5.2 and above installed on the server.
- Internet Information Services (IIS) components installed on the server.
- Refer to the *License Activation Starter* email for how to activate your license(s).

HTTP VS HTTPS

The default protocol for web access is HTTP, and this is what you will likely use when setting up and testing your new services *on your server*. For access from other locations, especially across the public internet, it is recommended to use the HTTPS protocol. You must have an SSL Certificate that is installed on your server to enable HTTPS protocol for your web service. In our Getting Started section below, it is assumed you are using HTTPS for your new server. If this is not the case, replace all instances of “HTTPS” in the examples with “HTTP”.

COMPLY WITH JSON SPECIFICATIONS

JSON (JavaScript Object Notation) is a lightweight data-interchange format that is easy to read and write and completely language independent but uses conventions that are familiar to programmers of the C-family of languages. Your software should be fully compliant with the JSON specifications according to ECMA-404.

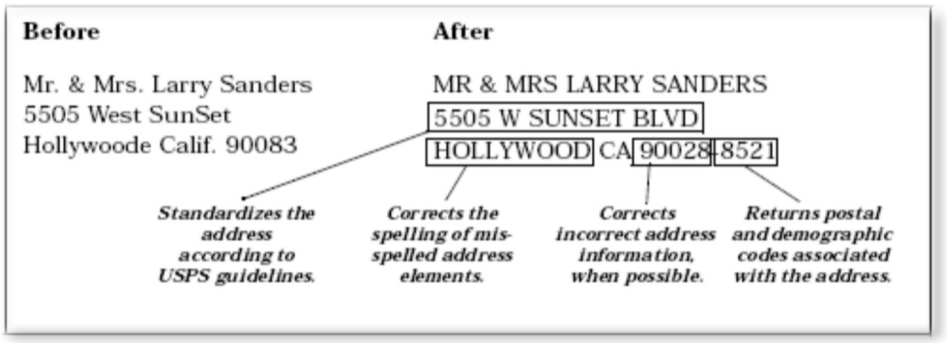
USE SECURE ENDPOINTS

Connections to the Verify REST service should be done securely using HTTPS/TLS and avoid using any plaintext endpoints.

INTRODUCING THE ACCUMAIL VERIFY SELF-HOSTED REST SERVICE PACKAGE

WHAT IS THE ACCUMAIL VERIFY SELF-HOSTED REST SERVICE PACKAGE?

The AccuMail Verify REST service takes addresses as input, checks to make sure they are deliverable, reformats them according to the U. S. Postal Service (USPS) CASS addressing standards, and appends important address codes, such as the ZIP+4 and carrier route codes. The following example demonstrates an address before and after it has been processed:



In addition to CASS address correction and coding, the AccuMail Verify Self-hosted REST service is available with an optional GeoCode module which provides location data (latitude, longitude and more) on addresses which have been processed through address correction first, and an optional RDI (Residential Delivery Indicator) add-on.

The Self-hosted REST service gives you a powerful real-time address verification tool for your website, intranet or other browser-based applications, such as CRM systems using Representational State Transfer (REST) for performance, reliability, and scalability.

Example of an AccuMail Verify Integration Into An Online Shopping Cart

FEATURES

Simple Integration – The REST service is designed for rapid deployment on your Web server, so you'll be up and running quickly.

Total Address Accuracy – Addresses are verified, standardized, and any missing elements such as ZIP+4 or CRRT Code are added to each record. The REST service instantly validates against the very latest postal database from the USPS.

Full Customization – Since the AccuMail Verify Self-hosted REST Service provides no user interface or web pages that are user-facing, you have complete control over the look and feel of your web forms for seamless integration within your site or application.

WHY USE THE ACCUMAIL VERIFY SELF-HOSTED REST SERVICE?

More Accurate Data – it costs much less to correct addresses at their point of entry than trying to fix issues that arise once bad addresses have entered the system.

Lower Mailing and Shipping Costs – cut undeliverable mail and packages and save on postage, printing and other costs, such as expensive Address Correction Penalties charged by some shipping carriers.

Increased Customer Satisfaction – lost or late deliveries due to inaccurate address data can damage your reputation and hurt your business. AccuMail Verify certifies that you get the right address every time and can even improve your delivery times. Plus, the entire user experience, will leave a positive impression on your website visitors, who can see that you care about getting things right.

Helps to Fight Fraud - verifying addresses prior to accepting orders through your website can help fight against fraudulent purchases.

Increased ROI – less returned mail, lower mailing and packaging costs, fewer address correction penalties from carriers, less waste, greater returns on your sales and marketing campaigns, faster customer payments and more.

You will enjoy the benefits of accurate data across your entire organization.

HOW DOES THE ACCUMAIL VERIFY SELF-HOSTED REST SERVICE WORK?

The Self-hosted REST service provides a bridge between your own application or website and the AccuMail address correction engine, a program that corrects and codes addresses. The REST service allows the developer to develop a routine that passes input data to your server for correction and coding and retrieves the corrected and coded result addresses along with other output information associated with the address.

THE USPS NATIONAL DATABASE

The USPS National Database is stored and updated on your server. It is a standardized and highly compressed version of the approximately 6-gigabytes of USPS data files containing virtually every deliverable address in the United States.

When you supply the Self-hosted REST service with an address, it is against the reference data in this database that the input address is compared. SmartSoft provides monthly updates that you can download and apply with minimal downtime to the users. These updates to the database are important and ensure that your addresses will be corrected and coded with the most current and accurate data available. Each data update will expire approximate 2.5 months after it is released.

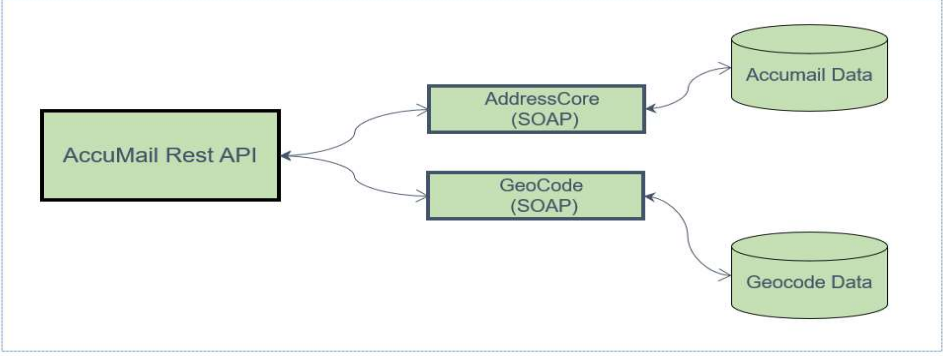
ASSUMPTIONS

- Root base URLs vary from environment to environment.
- End points are relative to the base URL.
- All REST service requests will return the results in JSON by default.
- REST service URLs are relative to the endpoint along with any query parameters.
- Unless otherwise mentioned, all GET APIs will have the following template fields in JSON response:
  - success – flag value of true/false to indicate success of the REST service. Please note, that even though the Http code might be a 2xx success code, the API might fail with one or more errors related to the Verification engine failure, Geocode result failure, coding exception error, etc.
  - message – Succeeded if no errors are returned. Failed request if there is an error.
  - errorCount – Returns the number of errors found.
  - errors - Collection (Dictionary) of errors.
  - failCount – Number of records**.**
  - resultCount – Number of records in the result.
  - result – Depending on the REST service result, it may be mono object, list of objects or container object.

INSTALLATION INSTRUCTIONS

OVERVIEW

This following contains installation instructions for the AccuMail Rest Server package. This solution is hosted on a server that is managed by your organization. The following figures illustrate the various server components:



SmartSoft AccuMail RestServer Package

Component	Description	Location
AccuMail Rest API	AccuMail Verify web service component	Server
SmartSoft GeoCode API	Web service hosting the SmartSoft Geocode Engine	Server
VerifyEngineV2 API	Web service hosting the SmartSoft Verify Engine	Server
AccuMail Data	Reference Data used by Verify Engine web service	Server
SmartSoft GeoCode Data	Reference Data used by SmartSoft Geocode Engine web service.	Server

SMARTSOFT FOLDER SETUP

Create *all* folder structures listed in this entire section on the hosting server.

- AccuMail and GeoCode data should be deployed to a folder under *C:\SmartSoft.* See the reference data sections following. *Note:* Application pool Identity user running the Web apps should have read/write permission to this folder. See the *Creating App Pools* section below.
- Log folder should be *C:\Smartsoft\Log*.

ACCUMAIL REFERENCE DATA

- Unzip the AccuMail Data installer (e.g. AccuMailAPIData.2021.04.11.39.zip) and run SmDataInstall.exe. The recommended install path is C:\SmartSoft\AccuMail\Data.

ACCUMAIL RDI DATA

- Unzip the AccuMail RDI installer (e.g. AccuMailRDIData.2021.07.11.47.zip) and run SmDataInstall.exe. The recommended install path is C:\SmartSoft\AccuMail\Data. Note: RDI data will only be fulfilled if the RDI option has been purchased.

ACCUMAIL GEOCODE REFERENCE DATA

- Unzip the GeoCode data installer (e.g. GeoCodeData.2021.05.03.10.04.zip) and run SmDataInstall.exe. The recommended install path is C:\SmartSoft\AccuMailGeoCode\Data. Note: RDI data will only be fulfilled if the GeoCode module has been purchased.

Note: Your web config entries for data install paths must match the actual install paths used.

LICENSE ACTIVATION

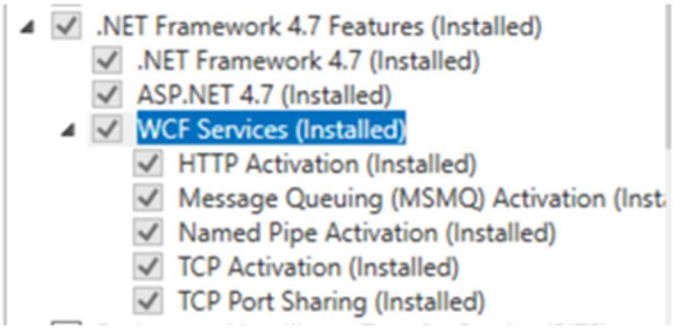
In addition to your email containing the links to the AccuMail Verify Self-hosted REST service package and this documentation, you should also receive a *License Activation Starter* email, containing everything needed to install and activate the license(s) for the modules you purchased with your package.

Please refer to the instructions in the *License Activation Starter* email to get started using your new product.

DEPLOYING ACCUMAIL VERIFY SELF-HOSTED REST

To begin the deployment of the service, IIS (Internet Information Services Manager) must be installed on your server. The following sections assume that this step has been done successfully and that IIS is running.

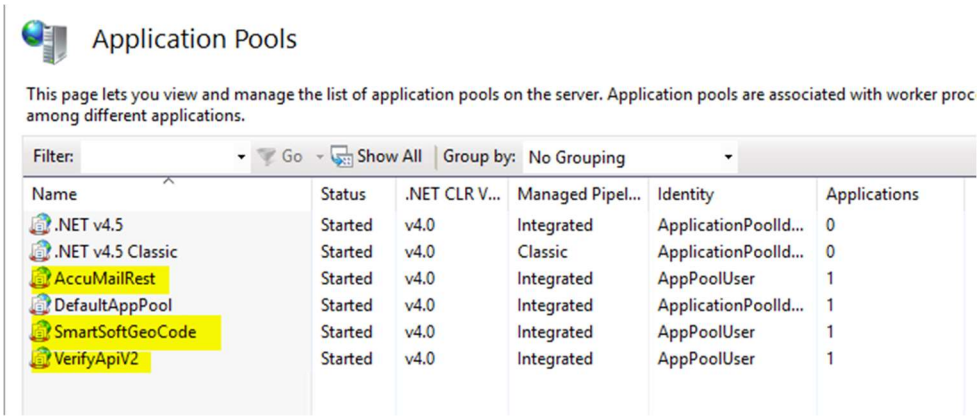
Additionally, please make sure that .NET 4.7 is installed on the server, including ALL WCF services (which is not the default):



CREATING IIS APP POOLS FOR ACCUMAIL SELF-HOSTED REST

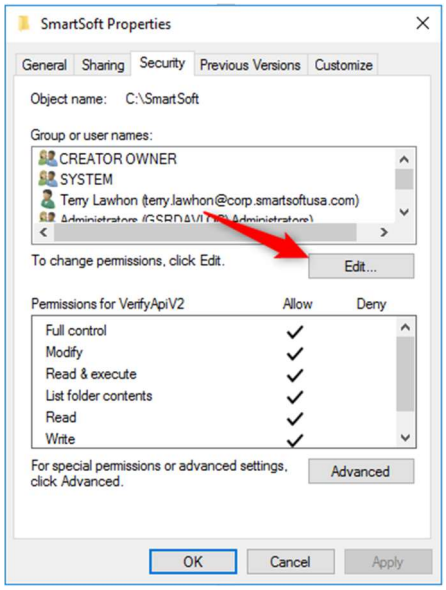
Create one Application Pool per web service application.

Example (with suggested App Pool names):

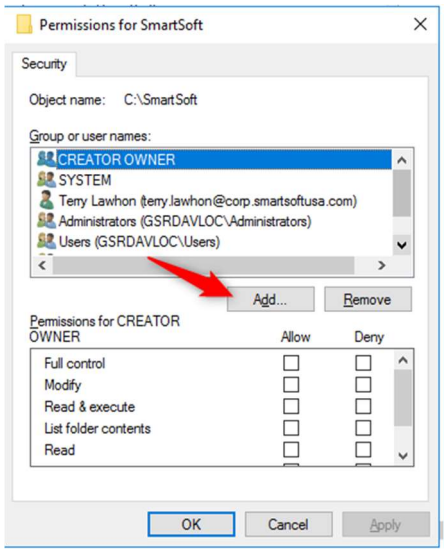


Ensure that the ApplicationPool Service account has read/write permission to both the Log and Reference data folders. The identity to give permissions to is normally related to the App Pool name you assigned to each service.

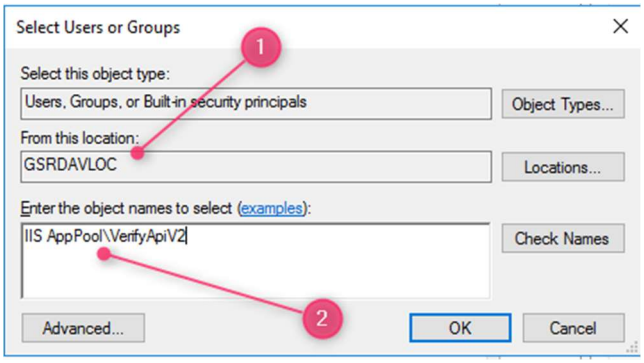
1. Assuming you used the recommended folder structure for data and log folders, you would right-click on the C:\SmartSoft folder in Windows Explorer and select Properties.
2. Click on the Security tab, and then click the Edit button:



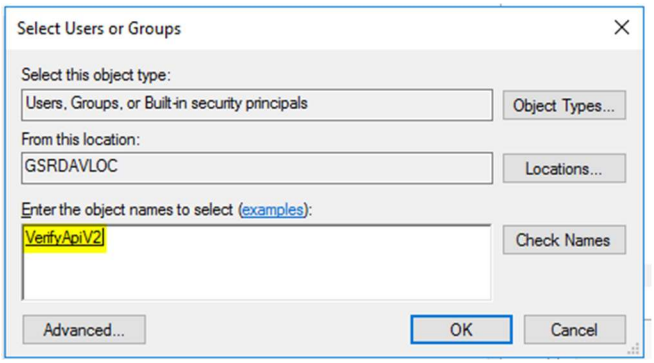
3. On the Permissions dialog, click Add:



4. On the Select Users or Groups dialog, first (①) make sure to select the local machine name for Location, then (②) enter the identity to add. The protocol for App Pool identity names is **IIS AppPool\<name>**, where **<name>** is the name you gave to the App Pool. Example:



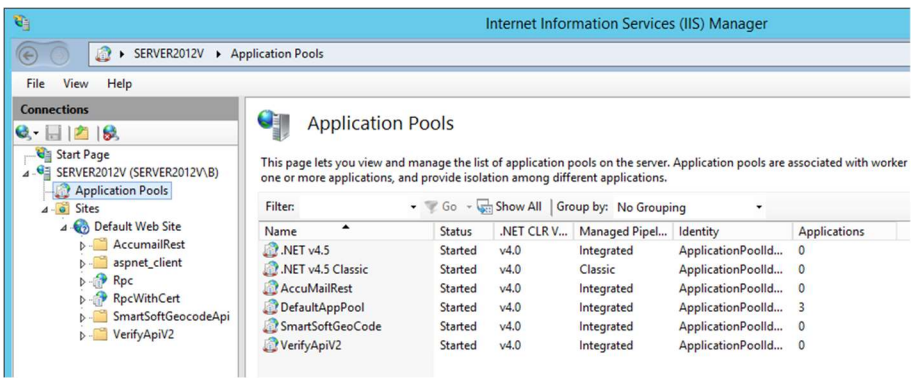
5. Click the Check Names button. If you entered the identity correctly, the entry will change to the actual App Pool identity, as shown below:



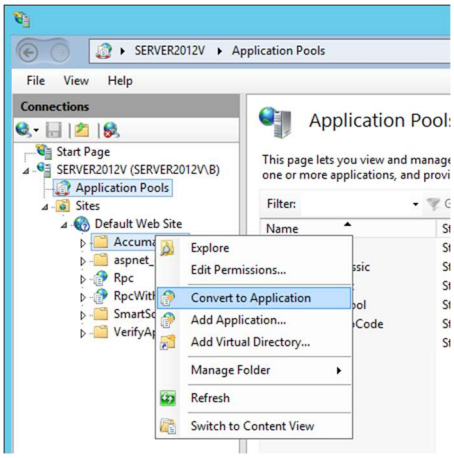
6. Each App Pool Identity you created in IIS should be given Full control access to the folders for your data and logs.

DEPLOYING WEB APPLICATIONS TO IIS.

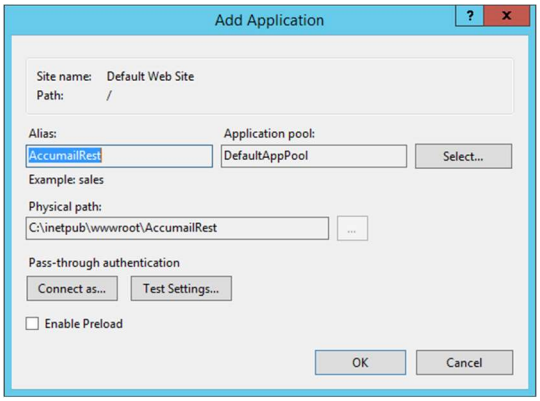
- Extract the entire contents of RestServerPackage.zip to C:\inetpub\wwwroot, making sure to preserve the folder structure in the archive.
- Repeat the following steps for *EACH* of the services purchased: (i.e. AccuMailRest; SmartSoftGeoCodeAPI; VerifyApiV2)
    - Open IIS and and drop down the main server list, then drop down Sites, and then the Default Web Site.
    - You should see folders for the contents you extracted from the package zip in the previous step (i.e. AccumailRest, VerifyApiV2, etc.).



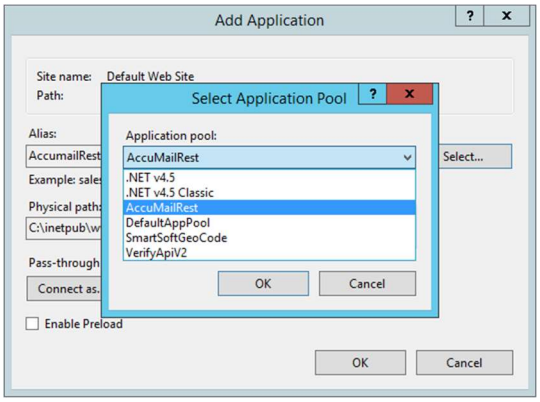
- Right-click on each of the service folders and select Convert to Application.



- On the Add Application dialog, select the appropriate App Pool that you created previously for each service.



- Click OK to select the App Pool, and then click OK to finish converting to an application.



- Repeat for each application being installed.

APPLICATION SETTINGS

The Application Settings for each installed service must be set to correctly reference your installation. View or edit your Application Settings by selecting the service in IIS and then double-clicking Application Settings.

VERIFYAPIV2

Application Setttings for VerifyApiV2 should be configured as follows:

Name	Contents	Notes
AccuMailDataFolder	Folder path - text	Location of AccuMail reference data. Usually - C:\SmartSoft\AccuMail\Data
AccuMail.LicenseFolder	License folder path - text	Location of License file for AddressCore. Usually - C:\SmartSoft \Licenses
LogFolder	Log folder path - text	Log folder path. Usually - C:\SmartSoft\Log
LogFilePrefix	Log file prefix - text	Rolling log filenames will be {prefix}yyyy-mm-dd.txt (Do not alter)
LogLevel	Log level - integer	Log level 1 - Error, 2 - Warning, 3 - Info, 4 - Debug (all levels)
AC.LogFilePath	Address core engine log file - text	Usually - {LogFolder}\{LogFilePrefix}ACLog.txt
AC.VerboseLog	Address core engine log verbosity - boolean	true/false. Use false unless directed by SmartSoft Support.

SMARTSOFTGEOCODEAPI

Application Settings for SmartSoftGeoCodeApi should be configured as follows:

Name	AppSetting	Notes
GeoCodeDataFolder	Folder path - text	Location of AccuMail GeoCode reference data. Usually - C:\SmartSoft\AccuMailGeoCode\Data
GeoCode.LicenseFolder	License folder path - text	Location of license file for GeoCode. Usually - C:\SmartSoft\Licenses
LogFolder	Log folder path - text	Log folder path Usually - C:\SmartSoft\Log
LogFilePrefix	Log file prefix - text	Rolling log filenames will be {prefix}yyyy-mm-dd.txt (Do not alter)
LogLevel	Log level - integer	Log level 1 - Error, 2 - Warning, 3 -Info, 4 - Debug (all levels)

ACCUMAILREST

- If you have one Address engine instance configure the setting **engineAPIUrls** to reference the VerifyEngineV2.svc file.
    - <add key="**engineAPIUrls**" value="http://localhost/VerifyApiV2/VerifyEngineV2.svc " />

- If you have more than one Address engine instance configure the setting **engineAPIUrls** to reference both the VerifyApiV2 and VerifyApiV2a URLs. Be sure to separate the URL's with a semi-colon.
    - <add key="**engineAPIUrls**" value="http://localhost/VerifyApiV2/VerifyEngineV2.svc;http://localhost/VerifyApiV2a/VerifyEngineV2.svc" />

- Configure the setting **geocodeAPIUrl** to reference the Geocode.svc file.
    - <add key="**geocodeAPIUrls**" value="http://localhost/SmartSoftGeocodeApi/Geocode.svc" />

- To validate that each purchased service is configured and licensed correctly, navigate via your browser to each of the following applicable URL's:

    - http://localhost/VerifyApiV2/VerifyEngineV2.svc
    - http://localhost/SmartSoftGeoCodeAPI/Geocode.svc
    - http://localhost/AccumailRest
    - http://localhost/AccumailRest/api/componentlicense

GETTING STARTED

This section provides high-level instructions on how to request an address verification using the Verify REST Service.

- Select an application to test the Self-hosted REST service from. There are many applications available for free that can be used to assist in testing your new REST service. Postman and SoapUI are good. There are also many REST client plugins/add-ons available for most Web browsers that can be used.

- This is what your starting URI should look like:

  https://YourServerHere:YourPortHere/AccumailRest/api/Address?

- Replace your server and your port with the proper values and then enter the URI into the application of your choice or as the URL into your Web browser. Success should return *true* if the authorization path is valid and the response should look something like the screenshot below. In this example, the response result fields will return blank strings since no input address was passed with the URL.


```
{
  "success": true,
  "message": "Succeeded",
  "errorCount": 0,
  "errors": null,
  "failCount": 0,
  "resultCount": 1,
  "result": {
    "destinationAddress": {
      "company": "",
      "street": "",
      "street2": "",
      "street3": "",
      "suite": "",
      "urbanization": "",
      "city": "",
      "state": "",
      "zip": "",
      "zipPlusFour": "",
      "zip9": "-"
```


- Replace your server and your port and pass a full address similar to the example below. The Verify REST service is transparent enough that the request parameters can be passed in any order. Edit the URI below with your server and your port and paste it into a Web browser or REST client application.

  https://YourServerHere:YourPortHere/AccumailRest/api/Address?company=Datatech+SmartSoft+Inc&city=Westlake+Village&showAddressComponent=true&showGeocode=true&showLocationDetails=true&standardizeStreets=true&state=CA&street=31111+Agoura+Road&Street2=Suite+250&street3=Room+1&useAlias=true&useMixedCase=false&zip=91361

- The results below display output of the response from the Verify REST Server. You have successfully looked up an address using the AccuMail Verify REST Service!


```
"success": true,
"message": "Succeeded",
"errorCount": 0,
"errors": null,
"failCount": 0,
"resultCount": 1,
"result": {
  "destinationAddress": {
    "company": "",
    "street": "31111 AGOURA RD STE 250 RM 1",
    "street2": "",
    "street3": "",
    "suite": "",
    "urbanization": "",
    "city": "WESTLAKE VILLAGE",
    "state": "CA",
    "zip": "91361",
    "zipPlusFour": "",
    "zip9": "91361-"
```


See REQUEST COMPONENTS for a full list and description of valid required fields and settings.

REQUESTED INPUT FIELDS

Detailed information on the input address fields passed to the REST service

Parameter	Type	Description
company	String	**Optional**. Retrieves the company name that was entered through the input *Company* field. This field when provided, results in greater +4 accuracy.
street	String	**Required**. Sets the *Street* field of the lookup record. Use this field type to pass all information relating to the primary street address, including the street name, house number, directional, and street suffix. You can also use this field type to pass all other acceptable forms of primary address information, such as PO Box numbers, rural route numbers, and highway contract numbers. In addition, suite and apartment information can be passed with primary address information through the *Street* field. For example, "123 Main St" and "123 Main St, Apt A" are both acceptable. Suite and apartment information can also be passed using the *Street* field.
street2	String	**Optional**. Sets the *Alternate Street* field of the lookup record. Use this field to pass alternative primary street information. If AccuMail Verify fails to code the input address using the information in the *Street* field described above, then it attempts to code the address using the alternate street information in this field, if any exists.
street3	String	**Optional**. Sets the *Alternate Street* field of the lookup record. Use this field to pass alternative primary street information. If AccuMail Verify fails to code the input address using the information in the *Street* field described above, then it attempts to code the address using the alternate street information in this field, if any exists.
city	String	**Optional** if *Zip* field is set, otherwise **Required**. Sets the *City* field of the lookup record. Use this field type to pass either an input city name or an entire input 'last line' string — that is, a formatted city/state/ZIP string, such as "Baltimore, MD 21234." If you do not use the *State* and *ZIP* fields described below, then AccuMail Verify treats the *City* field as a last line field.
state	String	**Optional** if the *Zip* field is set, otherwise **Required**. Sets the *State* field of the lookup record. Use this field type to pass input state information, such as a state name or its 2-digit USPS abbreviation. You can also pass input state information as part of a formatted last line string using the *City* field described above.
zip	String	**Optional** if the *City* and *State* fields are set, otherwise **Required**. Sets the *Zip* field of the lookup record. Use this field type to pass the input 5-digit ZIP Code. You can also pass input ZIP information as part of a formatted 'last line' string using the *City* field described above.
urbanization	String	Retrieves output urbanization information for Puerto Rican addresses.

ADDRESS CORRECTION SETTINGS (PASSED THROUGH REQUEST)

Detailed information on the input settings passed to the REST service

Parameter	Type	Description
standardizeStreets	Boolean	True or False Setting s*tandardizeStreets* to true will format the street, suite and street2 fields to USPS Address formatting standards.
showAddressComponent	Boolean	True or False This controls whether you want to have *addressComponent* information in the result or not.
showLocationDetails	Boolean	True or False This controls whether you want to have location details information in the result or not.
showGeocode	Boolean	True or False This controls whether you want to have Geocode information in the result. This is not part of standard service and must be purchased as an add-on.
useAlias	Boolean	True or False Uses Alias Street Names when an Alias is passed as the input. Setting this value to false forces AccuMail Verify to return the USPS preferred street name instead of the input Alias name.
useMixedCase	Boolean	True or False Returns information in mixed case. Setting to false returns information in upper case.

RESULT COMPONENTS

On success (HTTP status code 200 OK), single lookup results in 5 components

DESTINATION ADDRESS

Detailed information on the address output fields for *destinationAddress*

Parameter	Type	Description
company	String	Retrieves the company name that was entered through the input *Company* field, if any.
street	String	Retrieves the output street address. This includes relevant suite information, if such information was passed as input through the input *Street* field. *NOTE: If the information entered through the input Street field could not be matched, but the information entered through the input Alternate Street (street2) field could be matched, then the result from the matched street2 address is retrieved using the street field.*
street2	String	If AccuMail Verify fails to match the input address using the primary street information in the input *street* field, then it attempts to code the address using the alternate street information in the input *Alternate Street* (street2) field, if any exists. When both *street* and *street2* fields are used for input, whichever input string AccuMail Verify does not use to obtain a match is returned by this field type. In other words, this field type retrieves only bad or unused input street address information. Good address information is retrieved using the street field type described above.
urbanization	String	Retrieves output urbanization information for Puerto Rican addresses.
city	String	Retrieves the output city name.
state	String	Retrieves the output state abbreviation.
zip	String	Retrieves the output 5-digit ZIP Code.
zipPlusFour	String	Retrieves the output ZIP+4 Code. The ZIP+4 Code is the 4-digit extension only. You can retrieve the 5-digit ZIP Code using the *ZIP* field type described above.
Zip9	String	Retrieves the output 9 digit Zip Code which consists of the 5 digit zip plus a hyphen and the *zipPlusFour*.

ADDRESS COMPONENTS

Detailed information on the address components output fields

Parameter	Type	Description
lineOfTravelDirection	String	1-character sequence code (either *A* for Ascending or *D* for Descending) indicates whether delivery will be made in ascending or descending order.
lineOfTravel	String	Retrieves the output Line of Travel identifier consisting of a 4-digit number. The 4-digit number indicates the order in which delivery will be made within a given ZIP+4. Once a LOT code is appended to your data file records, you can use it to presort your mailings so that they qualify for Enhanced Carrier Route rates.
secondaryUnitDesignator	String	Retrieves the secondary unit designator for the output street address. For example, if the output street address is “123 Main St Apt 12,” the *secondaryUnitDesignator* is *Apt*.
houseNumber	String	Retrieves the house number for the output street address. For example, if the output street address is “123 Main St,” the house number is *123*.
unitNumber	String	Retrieves the unit number for the output street address. For example, if the output street address is “123 Main St Apt 12,” the unit number is *12*.
preDirectional	String	Retrieves the pre-directional for the output street address. For example, if the output street address is “123 E Main St,” the *preDirectional* designator is *E*.
streetName	String	Retrieves the street name for the output street address. For example, if the output street address is “123 Main St,” the *streetName* is *Main*.
streetSuffix	String	Retrieves the street suffix for the output street address. For example, if the output street address is “123 Main St,” the *streetSuffix* is *St*.
postDirectional	String	Retrieves the post-directional for the output street address. For example, if the output street address is “123 Main St N,” the *postDirectional* is *N*.
pmb	String	Retrieves the Private Mailbox number. Private companies offering mailbox rental services to individuals and businesses are considered commercial mail receiving agencies (CMRA).
deliveryPoint	String	The last two characters of the house number or PO Box number of an address, or in the case of high-rise addresses, the unit number.

LOCATION DETAILS

Detailed information on the address location results for *locationDetails*

Parameter	Type	Description
crrt	String	Retrieves the output carrier route code. This is a 4-digit code assigned to each address on a mail carrier's route.
rdi	String	Returns a *Y* if the address is residential.
county	String	Retrieves the output county name.
countyNumber	String	Retrieves the output county number. This is the 3-digit USPS code for the county in which the address resides.
stateFIPS	String	Retrieves the output State FIPS code. This is the 2-digit USPS code for the state in which the address resides.
congressionalDistrict	String	Retrieves the output congressional district code. This is a 2-digit identifier for the United States congressional district to which the input address belongs.

GEOCODE DETAILS

Detailed information on GeoCode results returned for *geocode*

Parameter	Type	Description
lat	String	Retrieved the Latitude in decimal degrees. An angular distance north or south from the equator of a point on the earth's surface, measured by the meridian of the point.
lng	String	Retrieved the Longitude in decimal degrees. An angular distance east or west on the earth's surface, measured by the angle contained between the meridian of a particular place and some prime meridian, as that of Greenwich, England and expressed either in degree or by some corresponding difference in time.
zipMatchFlag	String	Returns *Y* (True) when the geolocation returned is based on delivery post office for the 5-digit zip and not the actual street address.

VALIDATION DETAILS

Detailed information on address validation results returned

Parameter	Type	Description
lookupReturnCode	String	Retrieves the output error code. AccuMail Verify assigns an error code if the input record could not be found in the USPS National Database. This is a 2-byte code that identifies what was wrong with the input address and why AccuMail Verify could not match it.
lookupReturnDescription	String	Retrieve a text description of the *lookupReturnCode* field.
dpv	String	Retrieves the 6-byte DVP (Delivery Point Verification) match codes. The match codes indicate if the address is valid (and if not, why not), if the address is within a Commercial Mail Receiving Agency (CMRA), and if the address was flagged as a False Positive. 6-byte result string, as follows: **Byte 1** **blank** - The address was not coded by AccuMail Verify and therefore no DPV processing was performed. **Y** - All delivery point components of the address were DPV validated. D The address's building number was DPV validated but required unit-level information is missing. **S** - The address's building number was DPV validated, but the unit number is invalid. **N** - The address's building number is invalid. **Byte 2** **blank** - The address was not coded by AccuMail Verify and therefore no DPV processing was performed. **Y** - The address was found in the CMRA (Commercial Mail Receiving Agency) table. **N** - The address was not found in the CMRA (Commercial Mail Receiving Agency) table. **Byte 3** **blank** - The address was not coded by AccuMail Verify and therefore no DPV processing was performed. **Y** - The address was found in the False Positive table. This is a flag to determine whether a mailing list is being generated or created during validation. Creating a mailing list through DPV certification is not allowed by the USPS. **N** - The address was not found in the False Positive table. **Byte 4** **blank** - The address was not coded by AccuMail Verify and therefore no DPV processing was performed. **Y** - Address was found in the VACANT table. Meaning: A delivery point was active in the past, but is currently vacant (in most cases, unoccupied over 90 days) and not receiving delivery. **N** - The address was not found in the VACANT table

		Byte 5 **blank** - The address was not coded by AccuMail Verify and therefore no DPV processing was performed. **Y** - Address was found in the NOSTAT table. Meaning: This indicates the address is not receiving delivery, and the address is not counted as a possible delivery. These addresses are not receiving delivery because A) delivery has not been established; B) customer receives mail as a part of a drop; or C) the address is no longer a possible delivery because the carrier destroys or returns all of the mail. **N** - The address was not found in the NOSTAT table. **Byte 6** **blank** - The address was not coded by AccuMail Verify and therefore no DPV processing was performed. **Y** - P.O. Box Street-Style Address found. Meaning: This indicates which Street addresses in a database are actually Post-Office Box physical locations. **N** - The address is not a P.O. Box-Style address (PBSA).
dpvDescription	String	Retrieves a text description of the DPV field.
dpvFootNotes	String	Retrieves the 8-byte DPV (Delivery Point Verification) Footnotes string. This field returns up to four 2-character codes that supplement the DVP field codes (described above) providing additional information about the DPV match/mismatch. Up to four of the following 2-character codes will be returned: **AA** - The address was successfully coded by AccuMail Verify. **A1** - The address was not successfully coded by AccuMail Verify. **BB** - All components of the address were DPV validated. **CC** - The address's building number was DPV validated, but the unit number is invalid. **N1** - The address's building number was DPV validated but required unit -level information is missing. **M1** - A building number is missing for the input address. **M3** - The address's building number is invalid. **P1** - The input address is missing a required PO, RR, or HC Box number. **RR** - The input address was identified by DPV as a Commercial Mail Receiving Agency (CMRA). **R1** - The input address was identified by DPV as a Commercial Mail Receiving Agency (CMRA) but required unit-level information is missing.
lacs	String	A 1-character Locatable Address Conversion Service (LACS) code to identify records that have been converted to the LACS system. The LACS system is being used for many rural route addresses and city addresses that are being modified to city style addresses so that emergency vehicles, such as police cars and ambulances, can more easily find these locations.
corrections	String	Retrieves the output correction codes string. This string consists of single character codes that indicate the corrections AccuMail had to make to the input record.
correctionsText	String	Retrieves the verbose text meaning of the corrections code from the field listed above.

leftOvers	String	Retrieves any leftover information that was part of the input street address string but was not used for obtaining a match. Leftover information is input data that AccuMail Verify had to discard to correct the address. For example, if the input street address is “123 Main St Junk Here,” the output leftover information is *Junk Here*.
recordType	String	A specific single character code is return for the type of delivery address. **G** – General Delivery **M** – Multi-Carrier **P** – PO Box **R** – Rural Route/Contract **S** – Street
isRuralRouteDefault	String	Returns a *Y* for Rural Route default matched addresses.
isHighriseDefault	String	Returns a *Y* if the input address codes specifically to a Highrise address default. Inserting a valid secondary (suite, apt) will return a *N*.

RESULT CODES

These are codes given if your address could not be certified. These codes will be in the field *lookupReturnCode*, if you have it added to your text file.

LOOKUPRETURNCODE VALUES

Code	Description
Blank	The address has been successfully coded.
1	The address is coded but undeliverable (i.e. on side of street known to contain no houses). 1700 Thomas Ave Santa Barbara CA 93101
2	The ZIP code was not found, and the city and state cannot be used to determine a geographical area to search. 123 Main St Invalid City Name CA 00000
3	Coding would result in changing both ZIP and city. This is illegal for PO BOX and route type addresses. PO Box 12345 Gaviota CA 93117 NOTE: If the input city is invalid or blank, then it is legal to change both city and zip for these types.
4	The best match would result in too many suspicious changes. 123 State Ave Santa Barbara CA 93102 NOTE: AccuMail Verify considers some addressing errors as more serious than others. To prevent false matches, AccuMail Verify allows at most one of these serious errors. Here are the errors Smart Address Quality considerations: ▪ non-blank ZIP was changed ▪ an alias match was made which requires changing street to base name ▪ non-blank suffix was changed ▪ non-blank directional was changed (except for swapping of pre/post) ▪ more than 1 error in directional and suffix ▪ missing unit number (i.e. 123 foo st apt) ▪ missing SUD (i.e. 123 foo st 1) ▪ leftovers that must be discarded (i.e. 123 foo st PO Box 456) ▪ primary range errors (i.e. 123-4 foo st -> 1234 foo st)
5	The street was identified as an alias but was out of the range restricted for that alias. 4179 W Church St Pine Plains NY 12567 NOTE: The USPS indicates that some aliases apply only to a certain part of the base street. In the above example, the alias is limited to the range 3000-3816.
6	No street address was given. This occurs if the input street is blank AND a match cannot be made by swapping the firm or alternate street fields with the street field.

7 There are no street name matches in the given ZIP code or in any geographically-related ZIP code.
123 Invalid Street
Santa Barbara CA 93101

8 The street may contain superfluous components which cannot be discarded with confidence.
123 Anacapa St Unrecognized Extra Stuff
Santa Barbara CA 93101

9 The house number could not be matched.
12345 Anacapa St
Santa Barbara CA 93101

10 The best match was made to a ZIPMOVE record but was not an exact match.
2 Middle Ave
Cumberland Foreside ME 04110

11 A ZIPMOVE match was made but no exact match could be found in the new ZIP.
620 River Dell Rd
Breezewood PA 15533

12 The Early Warning System indicates that an exact match will become available in the next database update.

13 There are multiple matches with the same degree of confidence. This may indicate an inconsistency in the USPS data.
RR 3 Box 21
Alvin TX 77511

NOTE: This is often caused by overlapping ranges which have different ZIP+4s.

14 Incorrect suffix, directional, street name, or unit resulted in multiple matches with the same degree of confidence.
123 Islay St
Santa Barbara CA 93101

NOTE: This result indicates that information on the street line would have broken a tie between two or more matches.

15 Incorrect ZIP, city, or urbanization resulted in multiple matches with the same degree of confidence.
14 Cranch St
Quincy Center MA

NOTE: This result indicates that information on the last line would have broken a tie between two or more matches.

16 A corrected field was too long to fit into the supplied field.

17 Media Error. The database could not be read because of hardware or system problem.

18 Process Fault. Processing was interrupted due to serious program error. Please contact technical support.

19 DPV No Match. The address was found in the Zip+4 data, but not matched in the DPV reference.

20 A False Positive (USPS definition) was detected. Processing is suspended pending clarification

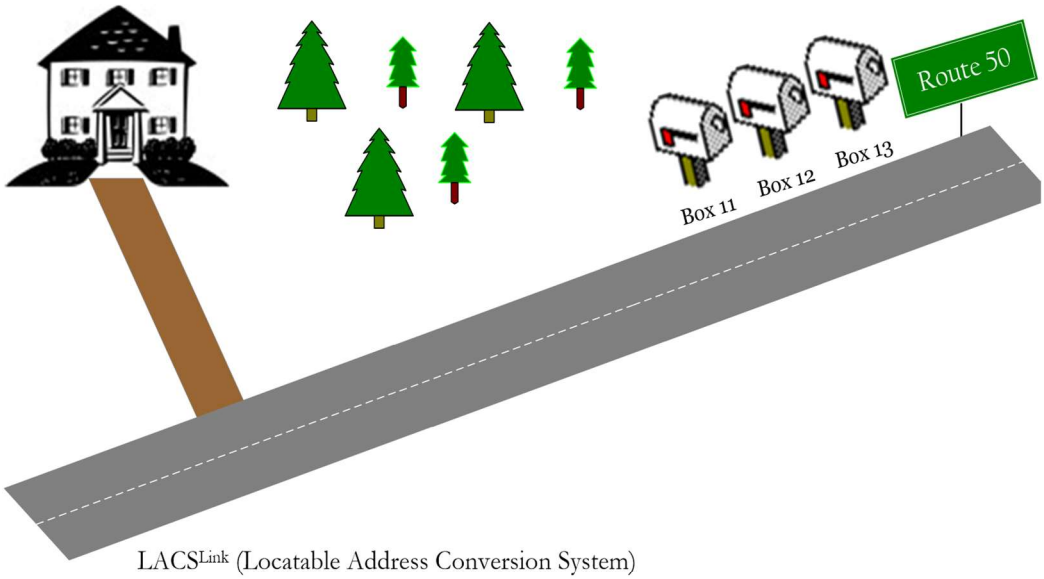
CORRECTION CODES

The correction codes are used to describe what was done to the address in the coding process. Each coded record will be assigned a string of one or more characters. The user may assign any size field to hold the correction codes. If the field is too short, then the codes will be truncated without error.

Code	Description
A	Normal street match
B	PO BOX match
C	Route type match
D	'Unique ZIP' match
E	'Small town' match
F	Alias match
G	'Highrise alternate' match
H	Firm match
I	Highrise match
J	Highrise default match
K	Route default match
L	Street name corrected
M	Street suffix corrected
N	Predirectional corrected
O	Postdirectional corrected
P	City corrected
Q	State corrected
R	ZIP corrected
S	Urbanization corrected
T	ZIP+4 corrected
U	House number corrected
V	Unit number corrected
W	Secondary unit designator corrected
X	Firm corrected
Y	Street swapped with firm
Z	Street swapped with alternate
0	Dual address changed to PO BOX
1	Dual address street match
2	Input city is not preferred but is acceptable
3	Street standardized
4	Unit not verified
5	Leftovers found
6	ZIPMOVE match
7	LACSLink match
8	SuiteLink match

DELIVERY POINT VALIDATION (DPV)

As the name implies, with DPV, AccuMail will validate that the USPS delivers to a given house number. Given a sample street as below, an input of 118 Main St, will not qualify for postal discounts. Only 110, 122, and 134 Main St will now get assigned a ZIP4.



LACS^{Link} (Locatable Address Conversion System)

Mailing Address Input: Route 50 Box 12

Converts to a Street Style Address: 30 Forrest Lane

Note the mail gets delivered fine to Route 50 Box 12, "Locatable" refers to the ability for emergency services to find the house.

The USPS is continuing its mission of automating the mail. Mail pieces that do not DPV validate cost them extra to deliver and therefore do not qualify for discounts.
A record that does not DPV confirm, can be matched by AccuMail for street standardization, city, state and ZIP correction, but is not assigned a ZIP+4 or Delivery Point information used for postage discounts. The AccuMail result field will contain 19 for this case.

DPV: TWO FIELDS (DPVCODE AND DPVFOOTNOTES)

DPVCode: This is commonly thought of as a Yes or No answer, which it generally is, but it contains more information as well. This field can be up to three characters in width. In general, only a Y, S, or D in the first position indicates the record was matched at some level of DPV and contains a ZIP4 for mailing.

DPVFootnotes: A DPV Footnotes code is returned for each mailing address that AccuMail produces a DPV Plus4 match. Refer to the DPV Footnotes Codes table on Page 22 for a description of each DPV Footnote code.

DPV CODES

The DPV field is five Characters (Bytes).

First Position

DPV Confirmation Indicator value and description:

Y = Address was DPV confirmed for both primary and (if present) secondary numbers.
D = Address was DPV confirmed for the primary number only, and Secondary number information was missing.
S = Address was DPV confirmed for the primary number only, and Secondary number information was present but unconfirmed.
N = Both Primary and (if present) Secondary number information failed to DPV Confirm.
Blank = Address not presented to DPV lookup table. (For example, the street or house number wasn't good enough to consider a match, there is nothing to search for.)

Second Position

DPV CMRA Indicator contains the results of the call to the DPV CMRA
(Commercial Mail Receiving Agency)

DPV CMRA Indicator value and description:

Y = Address was found in CMRA table.
N = Address was not found in CMRA table.
Blank = Address not presented to DPV. For example, a non-matched record and matched records that are in unique ZIP, military or General Delivery.

Third Position

DPV False Positive Indicator contains the results of the call to the DPV False Positive.

Note: This is a security test and processing will stop with a false positive hit. The USPS has security against generating lists using DPV. Certain seed records are considered *false positive* and require AccuMail to stop processing and notify the USPS. In this event, please send log.atz and falsepos.txt to technical support and you will be contacted to resolve the condition.

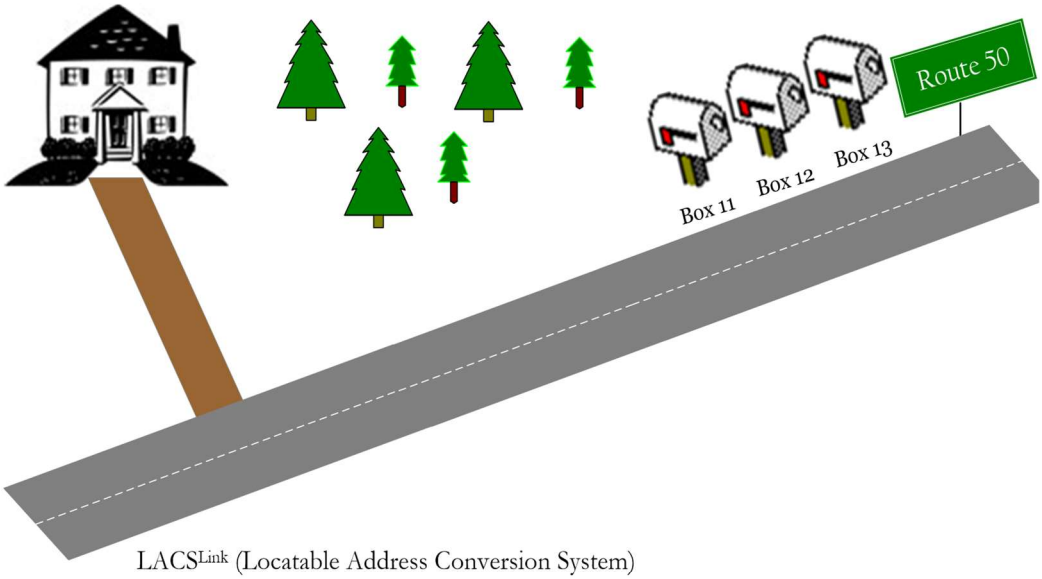
DPV False Positive Indicator value and description:

Y = Address was found in False Positive table.
N = Address was not found in False Positive table.
Blank = Address not presented to DPV

DPV FOOTNOTES CODES:

AA	Input Address Matched to the ZIP+4 file	P1	Input Address RR or HC Box number invalid
A1	Input Address Not Matched to the ZIP+4 file	P3	Input Address PO, RR, or HC Box number invalid
BB	Input Address Matched to DPV (all components)	RR	Input Address Matched to CMRA and PMB designator present (PMB 123 or #123)
CC	Input Address Primary Number Matched to DPV but Secondary Number not Matched *(present but invalid)*	R1	Input Address Matched to CMRA but PMB designator present (PMB 123 or #123)
N1	Input Address Primary Number Matched to DPV™ but Address Missing Secondary Number	F1	Input Address Matched to a Military Address
M1	Input Address Primary Number Missing	G1	Input Address Matched to a General Delivery Address
M3	Input Address Primary Number Invalid	U1	Input Address Matched to a Unique ZIP Code
On footnotes of F1, G1 and U1 move *Y* to the DPV return code and spaces to all other flags.			

LACS[Link] may convert a rural style address to a more standard street name and house number. **Locatable** is the key concept; namely, where is the house physically located (rather than a box on the road that the mail gets delivered to). Note this mostly applies to Rural Routes and Highway Contract but may apply to any style of input street address.



LACS[Link] (Locatable Address Conversion System)

Mailing Address Input: Route 50 Box 12

Converts to a Street Style Address: 30 Forrest Lane

Note the mail gets delivered fine to Route 50 Box 12, "Locatable" refers to the ability for emergency services to find the house.

Whether an address gets converted depends on the LACS[Link] data. There is no way to tell simply by looking at an address if it will convert. AccuMail will attempt to convert records that have an *L* in their LACS Flag from the ZIP4 Data. AccuMail will also attempt to convert certain rural style of addresses or streets in unique ZIP codes that fail to match in the ZIP4 data. If the ZIP4 data suggests that there may be a conversion, it likely will be made but is not guaranteed. If there is a conversion, the LACS field in this case will be set to *L*. The street field is updated to the converted street if a conversion can be made.

BASIC REST CODING EXAMPLES

This section provides basic coding samples in C#, JAVA, and Python.

C# SAMPLE

The following code sample issues a request and returns the entire response in a single string from C# Console application.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.IO;

namespace VerifyRESTCSharpSample
{
  class Program
  {
    static void Main(string[] args)
    {
      string result = null;
      string url =
      "https://api.smartsoftdq.com/Verify/api/Address?authorizationtoken=[YOUR-API-KEY]";

      // Sends the request to Verify REST Service
      HttpWebRequest request = WebRequest.Create(url) as HttpWebRequest;

      // Returns the response from the Verify REST Service
      HttpWebResponse response = request.GetResponse() as HttpWebResponse;
      {
        // Uses StreamReader to return the result as a single string
        StreamReader reader = new StreamReader(response.GetResponseStream());
        result = reader.ReadToEnd();
      }
    }
  }
}
```

PYTHON SAMPLE

The following code sample is like the C# sample as it returns the entire response as a string in a simple GET request.

Httplib2 Library Documentation:
http://httplib2.readthedocs.io/en/latest/

The httplib2 library can be downloaded from the link below:
https://pypi.python.org/pypi/httplib2#downloads

```
import httplib2

http = httplib2.Http()
response, content = http.request(https://api.smartsoftdq.com/Verify/api/Address?authorizationtoken=[YOUR-API-KEY]')
print content
```

JAVA SAMPLE

The following code sample returns the entire response as a string. This function takes the URL as string, issues a request and returns the entire response in a single string.

```
public static String httpGet(String urlStr) throws IOException
{
  URL url = new URL(urlStr);
  HttpURLConnection conn =
     (HttpURLConnection) url.openConnection();

  if (conn.getResponseCode() != 200) {
    throw new IOException(conn.getResponseMessage());
  }

  // Buffer the result into a string
  BufferedReader rd = new BufferedReader(
     new InputStreamReader(conn.getInputStream()));
  StringBuilder sb = new StringBuilder();
  String line;
  while ((line = rd.readLine()) != null) {
    sb.append(line);
  }
  rd.close();

  conn.disconnect();
  return sb.toString();
}
```

SECURITY PROTOCOL

SmartSoft provides clients with address changes for all United States Addresses via the United States Postal Service (USPS). The company and USPS maintain a strong commitment to privacy while providing addressing details to individuals who live within the continental United States.

In keeping with the USPS security regulations, clients are not able to update these addresses without first:

1. Identification: This is accomplished through SmartSoft, Inc. AccuMail Verify Web Service product license keys which are uniquely generated for SmartSoft clients, then verified during each Single Lookup job request.
2. Encrypted Communications: The exchange of said information is to occur over a secure and encrypted method of communication. SmartSoft accomplishes secure communication over HTTPS (Hypertext Transfer Protocol with the SSL/TLS protocol) to provide encrypted communications between your server and the Web service.
3. Intrusion Detection: SmartSoft takes the appropriate steps to ensure data is secure, from both internal and external sources.
4. Auditing: All accesses to the AccuMail Verify Web Service server are stored in a relational database including details related to time, client and activity.
5. Electronic Network Security: Electronic security and verification is controlled by a variety of industry-standard methods.
6. User specific Accounts and Passwords: SmartSoft clients pass through multiple levels of industry-standard security when processing each job request with the use of unique accounts and passwords.
7. Client Access Auditing: SmartSoft tracks certain metadata related to each job processed. Tracking usage allows SmartSoft, Inc to audit individual usage across accounts, licenses and products. This does not include tracking or durational storage of specific customer data beyond what is required for job processing.

ADDITIONAL HELP AND CONTACT DETAILS

For more information on any SmartSoft product please call us at 888.227.7221.

SmartSoft Inc.
30700 Russell Ranch Rd Ste 250
Westlake Village CA 91362-9507
888.227.7221 or 818.707.9300
Website: www.smartsoftDQ.com
Email: contact@smartsoftusa.com

Technical Support
Toll Free: 800.578.8324
Email: support@smartsoftusa.com

Accounting
800.361.1409 (from the US) or 514.332.0741
800.361.9673 (from Canada)