



DOTS Address Validation 2 - US

Developer's Guide

Note: This is an enhanced version of our DOTS Address Validation – US service. It has been optimized for speed, stability and throughput using the latest .NET technologies but is otherwise identical to the older service. Current clients are welcome to contact us for more information on upgrading. We will continue to support the older operation as well and a copy of the developers guide for that can be found here:

http://www.serviceobjects.com/devguides/DOTS_Address_Validation_US_1.pdf

Version 1.0
March 23, 2012
Jonas Shaefer

Table of Contents

Introduction	3
Web Service Structure	3
Operation Definitions	5
ValidateAddress	5
ValidateAddressWithExtras	7
ValidateAddressWithFragments	9
ValidateAddressWithDPV	11
GetPossibleMatches	14
ValidateAddressSingleLine	16
ValidateAddressType	18
ValidateCityStateZip	19
ParseAddress	20
ValidateAddressWithRDI	22
ValidateAddressWithSuiteLink	26
Error Codes	31
Integration / Frequently Asked Questions	32
Conclusion	36

Introduction

DOTS Address Validation 2 US (AV2) is a publicly available XML web service that provides parsed and corrected information about a physical US address. The service provides corrected information such as the correct street location and zip plus four code, along with parsed address tokens, such as the PMB box number, pre- and post-directionals, county and state codes, and much more.

AV2 can provide instant address verification and correction to websites or enhancement to contact lists. However, the output from AV2 must be considered carefully before the existence or non-existence of an address is decided.

Web Service Structure

Web services are methods that integrate with other applications via the web, and encapsulate tricky business logic. Web services are too large of a topic to cover in this document, but ServiceObjects has developed its web services to be as easy to integrate and as accessible as possible.

AV is a public XML web service that supports SOAP, POST and GET operations. Note that SOAP is done via POST, only with special XML markup in the post-body.

The host path, or physical location of the web service is here:

<http://trial.serviceobjects.com/av2/AddressValidate.asmx>

The location of the WSDL, or Web Service Definition Language document, is here:

<http://trial.serviceobjects.com/av2/AddressValidate.asmx?WSDL>

(This is also accessible via the “Service Definition” link.)

This XML is the definition of the web service, meaning its inputs, outputs, operations, and the like. Most likely, you will have another tool read this WSDL and make the operations available to you in your application. Whenever your utilities or IDE asks for a WSDL path to AV2, you can provide this one.

Every web service has *operations* that it offers to subscribers – methods that do different work and return different output. Examining the link above, you will notice several of these operations available, which are described in detail later on.

ValidateAddress – validates an address’s authenticity, and appends locational metadata about the address.

ValidateAddressWithExtras – like ValidateAddress, but accepts an Address2 field in the case that it may help with the address’s validation.

ValidateAddressWithFragments – like ValidateAddress, but returns parsed elements of the address’s street location.

GetPossibleMatches – like ValidateAddressWithFragments, but may return many possible matches.

ValidateAddressSingleLine – like ValidateAddressWithFragments, but takes all the address’s elements on the same line.

ValidateAddressType – Returns address type information about an address – whether it is a PO Box, street location, high-rise, etc.

ValidateCityStateZip – Returns whether the given city/state/zip combination match correctly.

ParseAddress – returns a corrected address’s parsed elements and fragments, even if the address did not validate. Much like ValidateAddressWithFragments, only guaranteed to return parsed address information.

ValidateAddressWithDPV – Returns parsed and validated address elements including Delivery Point Validation.

ValidateAddressWithRDI – Functions identically to ValidateAddressWithDPV except it also returns the “Residential Delivery Indicator” (RDI) flag. This operation is only available to active DOTS Address Validation - US subscribers through a special agreement. Please call Service Objects for more information about the RDI operation.

ValidateAddressWithSuiteLink – Functions identically to ValidateAddressWithDPV except it accepts a business name to potentially append Suite data if it’s missing.

Each of these operations will be described in detail later in this document.

Operation Definitions

This document defines the input, output and behavior of the web service operations in AV2. Each operation has its own unique behavior and output, although some of the operations are very similar.

Important Note!

Address Validation is a difficult task because the US Postal Service wants to protect residents from getting loads of junk mail. For this reason, the USPS either gives data in ranges (1 – 100 Main street) or, under very special and rare circumstances, actual street locations.

Users who are looking to validate whether a specific street location is deliverable by the USPS or existing building (delivery-point validation) will need to use our DPV operation, as standard AV2 will only validate addresses based on ranges, not deliverable status.

You should examine your own business needs first before exploring what AV2 can provide. Different operations may be necessary based on the information you have, and the granularity of data you need.

ValidateAddress

This is the basic operation for validating and correcting a US address. This operation takes a standard US address (Address, City, State, Zip) and will try to verify its authenticity and return validated address elements. It cannot always find the location, especially when it comes to empty lots or new construction areas.

This operation will also return locational metadata, such as county name/code, congress code, and barcode digits, for the given address. This data is valuable for tracking demographics and statistical analysis.

This operation requires the Address value, and either City and State, or the Zip code. Providing all inputs is recommended, because it helps the validation proceed if some of the elements are malformed.

ValidateAddress Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, "123 Main Street".

City	String	The city of the address to validate. For example, "New York". The city isn't required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, "NY". This does not need to be contracted, full state names will work as well. The state isn't required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateAddress Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line
City	String	Varies	The corrected City
State	String	Varies	The corrected State
Zip	String	Varies	The corrected zip plus four
Address2	String	Varies	The corrected Address2, if applicable. Rarely appears in ValidateAddress outputs.
BarcodeDigits	String	12-digit number	The barcode numbers that would be applied to a parcel that USPS would deliver.
CarrierRoute	String	Varies	4 digits: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the corrected address.
CountyName	String	Varies	The county name of the corrected address.
CountyCode	String	Varies	The county code of the corrected address.
Fragment	String	Varies	The apartment, suite, or box number associated with the corrected address. The ValidateAddressWithFragments operation is probably more useful for these types of elements.

Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See “Error Codes” below.
Error – Location	String	Always null	Deprecated, no longer used.

ValidateAddressWithExtras

This operation is almost exactly like ValidateAddress, but rather than taking only an Address1 (Address) element, it also accepts an Address2. Typically, AV will only use Address2 in the case that Address1 is a nonsensical element (e.g., “Smith Building”.) However, if the Address2 is an element that does not assist with the validation of an address, it will be passed through to the output without any modifications made to it. For example, an Address2 of “Next to the stairs” does not assist with validation, but would be necessary for the delivery of packages to that address.

For operations that do not take Address2, you can append a comma, then a space, then your Address2 to the Address line, and it will be processed as the Address2. For example, if you had an Address like:

123 Main Street
 Building C
 Anytown, CA 99999

You could run this through ValidateAddress by passing elements as so:

Address: 123 Main Street, Building C
 City: Anytown
 State: CA
 Zip: 99999

ValidateAddressWithExtras Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, “123 Main Street”.
Address2	String	The Address2 of the address to validate. This will only be used in particular situations where the Address1 does not help in validating the address.

City	String	The city of the address to validate. For example, "New York". The city isn't required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, "NY". This does not need to be contracted, full state names will work as well. The state isn't required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateAddressWithExtras Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line
City	String	Varies	The corrected City
State	String	Varies	The corrected State
Zip	String	Varies	The corrected zip plus four
Address2	String	Varies	The corrected Address2. Often, if the Address2 is not part of the "official" address, it will be passed through to the output without any tampering.
BarcodeDigits	String	12-digit number	The barcode numbers that would be applied to a parcel that USPS would deliver.
CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the corrected address.
CountyName	String	Varies	The county name of the corrected address.
CountyCode	String	Varies	The county code of the corrected address.
Fragment	String	Varies	The apartment, suite, or box number associated with the corrected address. The

			ValidateAddressWithFragments operation is probably more useful for these types of elements.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See “Error Codes” below.
Error -- Location	String	Always null	Deprecated, no longer used.

ValidateAddressWithFragments

This operation is almost exactly like ValidateAddress, but instead returns all the parsed fragment tokens from the validated address.

ValidateAddressWithFragments Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, “123 Main Street”.
City	String	The city of the address to validate. For example, “New York”. The city isn’t required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, “NY”. This does not need to be contracted, full state names will work as well. The state isn’t required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn’t required, but if one is not provided, the City and State are required.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateAddressWithFragments Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line 1.
City	String	Varies	The corrected city name.
State	String	Varies	The corrected state name.

Zip	String	Varies	The corrected zip code + 4.
Address2	String	Varies	The corrected Address line 2.
BarcodeDigits	String	Varies	The post office delivery barcode digits.
CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the given address.
CountyCode	String	Varies	The county code of the given address.
CountyName	String	Varies	The name of the county in which the given address lies.
Fragment	String	Varies	The parsed "Fragment" box, apartment or unit number. Same as FragmentPMBNumber.
FragmentHouse	String	Varies	The parsed house number of the given address.
FragmentPMBNumber	String	Varies	The parsed apartment, box, unit, etc. number of the given address.
FragmentPMBPrefix	String	Varies	The parsed type of the apartment, box, unit, etc. For example, "APT" or "BOX".
FragmentPostDir	String	Varies	The parsed post-directional of the address's street. "West" in "North Main St West".
FragmentPreDir	String	Varies	The parsed pre-directional of the address's street. "North" in "North Main St West".
FragmentStreet	String	Varies	The parsed name of the street in the given address. "Main" in "North Main St West".
FragmentSuffix	String	Varies	The parsed suffix of the street in the given address. "St" in "North Main St West".
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See "Error Codes" below.
Error – Location	String	Always null	Deprecated, no longer used.

ValidateAddressWithDPV

Returns parsed and validated address elements including Delivery Point Validation.

ValidateAddressWithDPV Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, "123 Main Street".
Address2	String	The Address2 of the address to validate. This will only be used in particular situations where the Address1 does not help in validating the address.
City	String	The city of the address to validate. For example, "New York". The city isn't required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, "NY". This does not need to be contracted, full state names will work as well. The state isn't required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey*	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateAddressWithDPV Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line 1.
City	String	Varies	The corrected city name.
State	String	Varies	The corrected state name.
Zip	String	Varies	The corrected zip code + 4.
Address2	String	Varies	The corrected Address line 2.
BarcodeDigits	String	Varies	The post office delivery barcode digits.
CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3

			for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the given address.
CountyCode	String	Varies	The county code of the given address.
CountyName	String	Varies	The name of the county in which the given address lies.
Fragment	String	Varies	The parsed "Fragment" box, apartment or unit number. Same as FragmentPMBNumber.
FragmentHouse	String	Varies	The parsed house number of the given address.
FragmentPMBNumber	String	Varies	The parsed apartment, box, unit, etc. number of the given address.
FragmentPMBPrefix	String	Varies	The parsed type of the apartment, box, unit, etc. For example, "APT" or "BOX".
FragmentPostDir	String	Varies	The parsed post-directional of the address's street. "West" in "North Main St West".
FragmentPreDir	String	Varies	The parsed pre-directional of the address's street. "North" in "North Main St West".
FragmentStreet	String	Varies	The parsed name of the street in the given address. "Main" in "North Main St West".
FragmentSuffix	String	Varies	The parsed suffix of the street in the given address. "St" in "North Main St West".
DPV*	String	1-8	Number that correlates to a DPV result.
DPVDesc*	String	Varies	Explains DPV result.
DPVNotes*	String	1-14	Number that correlates to DPV notes description.
DPVNotesDesc*	String	Varies	Details about the DPV result.
Corrections*	String	1-41, 43	Number that correlates to a Corrections Description.
CorrectionsDesc*	String	Varies	Description of what was corrected in an address.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.

Error – Number	String	1, 2, 3, 4, 5	See “Error Codes” below.
Error – Location	String	Always null	Deprecated, no longer used.

*Possible values/descriptions listed below.

DPV Codes

DPV - DPV Desc:
<p>"1", "Yes, the input record is a valid mailing address"</p> <p>"2", "No, the input record is not in the DPV database of valid mailing addresses"</p> <p>"3", "The apartment or rural route box number is not valid, although the house number or rural route is valid"</p> <p>"4", "The input record is a valid mailing address, but is missing the apartment or rural route box number"</p> <p>"5", "Internal error. Please contact customer service"</p> <p>"6", "Internal error. Please contact customer service"</p> <p>"7", "Internal error. Please contact customer service"</p> <p>"8", "Internal error. Please contact customer service"</p>

DPVNotes - DPVNotesDesc:
<p>"1", "The input address matched the ZIP+4 record"</p> <p>"2", "The input address did not match the ZIP+4 record"</p> <p>"3", "The input address matched the DPV record"</p> <p>"4", "The input address primary matched DPV but the secondary did not"</p> <p>"5", "The input address is a military APO/FPO address"</p> <p>"6", "The input address is a general delivery address"</p> <p>"7", "The input address is a building name, and the primary is missing"</p> <p>"8", "The input address matched a DPV high-rise but a secondary was not input"</p> <p>"9", "The input address primary is invalid"</p> <p>"10", "The input address is missing a PO box, rural route, or highway contract box number"</p> <p>"11", "The input address (PO box, rural route, or highway contract) primary is invalid"</p> <p>"12", "The input address specified a PMB and matched a CMRA"</p> <p>"13", "The input address did not include a PMB and matched a CMRA"</p> <p>"14", "The input address matched and retained a unique ZIP code"</p>

Corrections - CorrectionsDesc:
<p>"1", "State not found"</p>

"2", "City not found"
"3", "Street not found"
"4", "Address not found"
"5", "Uncomputable +4 range"
"6", "+4 unavailable"
"7", "Multiple streets match"
"8", "Multiple addresses match"
"9", "Time ran out"
"10", "Output too long"
"11", "Unused and reserved"
"12", "Company phonetic match used"
"13", "First company match used"
"14", "State determined from city"
"15", "State determined from ZIP"
"16", "City phonetic match used"
"17", "City determined from ZIP"
"18", "Acceptable city name used"
"19", "Street phonetic match used"
"20", "Predirection dropped"
"21", "Predirection added"
"22", "Suffix dropped"
"23", "Suffix added"
"24", "Postdirection dropped"
"25", "Postdirection added"
"26", "Predirection standardized"
"27", "Street standardized"
"28", "Suffix standardized"
"29", "Postdirection standardized"
"30", "Street missing"
"31", "Alternate address used"
"32", "Unit standardized"
"33", "Unit missing"
"34", "Unit not found"
"35", "Unit unverified"
"36", "Rural box not found"
"37", "Leftovers found"
"38", "ZIP not found"
"39", "ZIP changed"
"40", "+4 changed"
"41", "Route changed"
"43", "Directional or Suffix is either missing or unused"

GetPossibleMatches

This operation is almost exactly like `ValidateAddressWithFragments`, but may return multiple address with fragments.

GetPossibleMatches Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, "123 Main Street".
Address2	String	The Address2 of the address to validate. This will only be used in particular situations where the Address1 does not help in validating the address.
City	String	The city of the address to validate. For example, "New York". The city isn't required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, "NY". This does not need to be contracted, full state names will work as well. The state isn't required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

GetPossibleMatches Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line 1.
City	String	Varies	The corrected city name.
State	String	Varies	The corrected state name.
Zip	String	Varies	The corrected zip code + 4.
Address2	String	Varies	The corrected Address line 2.
BarcodeDigits	String	Varies	The post office delivery barcode digits.
CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the given address.

CountyCode	String	Varies	The county code of the given address.
CountyName	String	Varies	The name of the county in which the given address lies.
Fragment	String	Varies	The parsed “Fragment” box, apartment or unit number. Same as FragmentPMBNumber.
FragmentHouse	String	Varies	The parsed house number of the given address.
FragmentPMBNumber	String	Varies	The parsed apartment, box, unit, etc. number of the given address.
FragmentPMBPrefix	String	Varies	The parsed type of the apartment, box, unit, etc. For example, “APT” or “BOX”.
FragmentPostDir	String	Varies	The parsed post-directional of the address’s street. “West” in “North Main St West”.
FragmentPreDir	String	Varies	The parsed pre-directional of the address’s street. “North” in “North Main St West”.
FragmentStreet	String	Varies	The parsed name of the street in the given address. “Main” in “North Main St West”.
FragmentSuffix	String	Varies	The parsed suffix of the street in the given address. “St” in “North Main St West”.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See “Error Codes” below.
Error – Location	String	Always null	Deprecated, no longer used.

*note: output node ArrayOfFragmentAddress may contain many FragmentAddress nodes which contain the outputs described above.

ValidateAddressSingleLine

Often, users have addresses as a single line of data, which may or may not be separated by commas, slashes, spaces, etc. For this reason,

ValidateAddressSingleLine behaves almost exactly like ValidateAddressWithFragments, but takes all of its input as a single field.

ValidateAddressSingleLine Inputs

Name	Type	Description
Address	String	Entire address to validate. For example, "123 Main Street Anytown CA 99999".
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateAddressSingleLine Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line 1.
City	String	Varies	The corrected city name.
State	String	Varies	The corrected state name.
Zip	String	Varies	The corrected zip code + 4.
Address2	String	Varies	The corrected Address line 2.
BarcodeDigits	String	Varies	The post office delivery barcode digits.
CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the given address.
CountyCode	String	Varies	The county code of the given address.
CountyName	String	Varies	The name of the county in which the given address lies.
Fragment	String	Varies	The parsed "Fragment" box, apartment or unit number. Same as FragmentPMBNumber.
FragmentHouse	String	Varies	The parsed house number of the given address.
FragmentPMBNumber	String	Varies	The parsed apartment, box, unit, etc. number of the given address.
FragmentPMBPrefix	String	Varies	The parsed type of the

			apartment, box, unit, etc. For example, "APT" or "BOX".
FragmentPostDir	String	Varies	The parsed post-directional of the address's street. "West" in "North Main St West".
FragmentPreDir	String	Varies	The parsed pre-directional of the address's street. "North" in "North Main St West".
FragmentStreet	String	Varies	The parsed name of the street in the given address. "Main" in "North Main St West".
FragmentSuffix	String	Varies	The parsed suffix of the street in the given address. "St" in "North Main St West".
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See "Error Codes" below.
Error – Location	String	Always null	Deprecated, no longer used.

ValidateAddressType

This operation returns the validated address's type. Possible types are:

- S – Street address
- P – Post office box
- R – Rural route or highway contract
- H – High rise apartment or office building
- F – Firm or business
- G – General delivery

ValidateAddressType Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, "123 Main Street".
Address2	String	Address line 2 of the address to validate.
City	String	The city of the address to validate. For example, "New York". The city isn't required, but if one is not provided, the Zip code is required.

State	String	The state of the address to validate. For example, "NY". This does not need to be contracted, full state names will work as well. The state isn't required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateAddressType Outputs

Name	Type	Values	Description
AddressType	String	S, P, R, H, F, G	The code of the Address Type for the given address.
AddressTypeDescription	String	Varies	A description of the address type code.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See "Error Codes" below.
Error -- Location	String	Always null	Deprecated, no longer used.

ValidateCityStateZip

This operation will validate that a given city-state-zip validate together properly. The inputs can be marginally incorrect, and this operation will correct them. For instance, a combination with a valid city, slightly misspelled state, and totally incorrect zip code will be corrected to a valid city – state – zip code combination. Note that in this case, the corrected zip code will be the "standard" or most common zip code for that city/state.

ValidateCityStateZip Inputs

Name	Type	Description
City	String	The city to validate.

State	String	The state to validate.
PostalCode	String	The zip code to validate.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateCityStateZip Outputs

Name	Type	Values	Description
City	String	Varies	The corrected city name.
State	String	Varies	The corrected state name.
PostalCode	String	Varies	The corrected zip code. The standard zip code for the city and state is used if it was corrected.
PostalCodeFound	String	True or False	Whether a corrected standard zip code could be found
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See “Error Codes” below.
Error -- Location	String	Always null	Deprecated, no longer used.

ParseAddress

This operation will parse the corrected address out into logical fragment elements, much like ValidateAddressWithFragments, but will return these parsed fragments even if the given address did not validate properly.

ParseAddress Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, “123 Main Street”.
City	String	The city of the address to validate. For example, “New York”. The city isn’t required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, “NY”. This does not need to be contracted, full state names will work as well. The state isn’t required, but if one is

		not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ParseAddress Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected or parsed Address line 1.
City	String	Varies	The corrected or parsed city name.
State	String	Varies	The corrected or parsed state name.
Zip	String	Varies	The corrected or parsed zip code + 4.
Address2	String	Varies	The corrected or parsed Address line 2.
BarcodeDigits	String	Varies	The post office delivery barcode digits, if address was validated.
CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the given address, if address was validated.
CountyCode	String	Varies	The county code of the given address, if address was validated.
CountyName	String	Varies	The name of the county in which the given address lies, if address was validated.
Fragment	String	Varies	The parsed "Fragment" box, apartment or unit number. Same as FragmentPMBNumber.
FragmentHouse	String	Varies	The parsed house number of the given address.
FragmentPMBNumber	String	Varies	The parsed apartment, box, unit, etc. number of the given

			address.
FragmentPMBPrefix	String	Varies	The parsed type of the apartment, box, unit, etc. For example, "APT" or "BOX".
FragmentPostDir	String	Varies	The parsed post-directional of the address's street. "West" in "North Main St West".
FragmentPreDir	String	Varies	The parsed pre-directional of the address's street. "North" in "North Main St West".
FragmentStreet	String	Varies	The parsed name of the street in the given address. "Main" in "North Main St West".
FragmentSuffix	String	Varies	The parsed suffix of the street in the given address. "St" in "North Main St West".
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See "Error Codes" below.
Error – Location	String	Always null	Deprecated, no longer used.

ValidateAddressWithRDI

Identical to the ValidateAddressWithDPV operation but also returns RDI (Residential Delivery Indicator) data in the form of the "IsResidential" output node. Returns parsed and validated address elements including Delivery Point Validation.

This operation is only available to DOTS Address Validation - US subscribers through a special agreement. Please contact Service Objects at (805) 963-1700 or by emailing the sales department at sales@serviceobjects.com for more information about Residential Delivery Indicator.

ValidateAddressWithRDI Inputs

Name	Type	Description
Address	String	Address line of the address to validate. For example, "123 Main Street".
Address2	String	The Address2 of the address to validate. This will only be used in particular situations where the Address1 does not help in validating the address.
City	String	The city of the address to validate. For example, "New York". The city isn't required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, "NY". This does not need to be contracted, full state names will work as well. The state isn't required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey*	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

*this operation requires a special agreement with Service Objects.

ValidateAddressWithRDI Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line 1.
City	String	Varies	The corrected city name.
State	String	Varies	The corrected state name.
Zip	String	Varies	The corrected zip code + 4.
Address2	String	Varies	The corrected Address line 2.
IsResidential	String	"true" or "false"	Residential Delivery Indicator flag. If "true", address is a residence. If "false", it's a business.
BarcodeDigits	String	Varies	The post office delivery barcode digits.
CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the

			given address.
CountyCode	String	Varies	The county code of the given address.
CountyName	String	Varies	The name of the county in which the given address lies.
Fragment	String	Varies	The parsed "Fragment" box, apartment or unit number. Same as FragmentPMBNumber.
FragmentHouse	String	Varies	The parsed house number of the given address.
FragmentPMBNumber	String	Varies	The parsed apartment, box, unit, etc. number of the given address.
FragmentPMBPrefix	String	Varies	The parsed type of the apartment, box, unit, etc. For example, "APT" or "BOX".
FragmentPostDir	String	Varies	The parsed post-directional of the address's street. "West" in "North Main St West".
FragmentPreDir	String	Varies	The parsed pre-directional of the address's street. "North" in "North Main St West".
FragmentStreet	String	Varies	The parsed name of the street in the given address. "Main" in "North Main St West".
FragmentSuffix	String	Varies	The parsed suffix of the street in the given address. "St" in "North Main St West".
DPV*	String	1-8	Number that correlates to a DPV result.
DPVDesc*	String	Varies	Explains DPV result.
DPVNotes*	String	1-14	Number that correlates to DPV notes description.
DPVNotesDesc*	String	Varies	Details about the DPV result.
Corrections*	String	1-41, 43	Number that correlates to a Corrections Description.
CorrectionsDesc*	String	Varies	Description of what was corrected in an address.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See "Error Codes" below.
Error – Location	String	Always null	Deprecated, no longer used.

*Possible values/descriptions listed below.

DPV Codes

DPV - DPV Desc:
"1", "Yes, the input record is a valid mailing address" "2", "No, the input record is not in the DPV database of valid mailing addresses" "3", "The apartment or rural route box number is not valid, although the house number or rural route is valid" "4", "The input record is a valid mailing address, but is missing the apartment or rural route box number" "5", "Internal error. Please contact customer service" "6", "Internal error. Please contact customer service" "7", "Internal error. Please contact customer service" "8", "Internal error. Please contact customer service"

DPVNotes - DPVNotesDesc:
"1", "The input address matched the ZIP+4 record" "2", "The input address did not match the ZIP+4 record" "3", "The input address matched the DPV record" "4", "The input address primary matched DPV but the secondary did not" "5", "The input address is a military APO/FPO address" "6", "The input address is a general delivery address" "7", "The input address is a building name, and the primary is missing" "8", "The input address matched a DPV high-rise but a secondary was not input" "9", "The input address primary is invalid" "10", "The input address is missing a PO box, rural route, or highway contract box number" "11", "The input address (PO box, rural route, or highway contract) primary is invalid" "12", "The input address specified a PMB and matched a CMRA" "13", "The input address did not include a PMB and matched a CMRA" "14", "The input address matched and retained a unique ZIP code"

Corrections - CorrectionsDesc:
"1", "State not found" "2", "City not found" "3", "Street not found" "4", "Address not found" "5", "Uncomputable +4 range"

"6", "+4 unavailable"
"7", "Multiple streets match"
"8", "Multiple addresses match"
"9", "Time ran out"
"10", "Output too long"
"11", "Unused and reserved"
"12", "Company phonetic match used"
"13", "First company match used"
"14", "State determined from city"
"15", "State determined from ZIP"
"16", "City phonetic match used"
"17", "City determined from ZIP"
"18", "Acceptable city name used"
"19", "Street phonetic match used"
"20", "Predirection dropped"
"21", "Predirection added"
"22", "Suffix dropped"
"23", "Suffix added"
"24", "Postdirection dropped"
"25", "Postdirection added"
"26", "Predirection standardized"
"27", "Street standardized"
"28", "Suffix standardized"
"29", "Postdirection standardized"
"30", "Street missing"
"31", "Alternate address used"
"32", "Unit standardized"
"33", "Unit missing"
"34", "Unit not found"
"35", "Unit unverified"
"36", "Rural box not found"
"37", "Leftovers found"
"38", "ZIP not found"
"39", "ZIP changed"
"40", "+4 changed"
"41", "Route changed"
"43", "Directional or Suffix is either missing or unused"

ValidateAddressWithSuiteLink

Identical to the ValidateAddressWithDPV operation but also accepts a business name in order to attempt to append Suite data. Also returns parsed and validated address elements including Delivery Point Validation.

ValidateAddressWithSuiteLink Inputs

Name	Type	Description
BusinessName	String	Name of business associated with this address. Used to append Suite data
Address	String	Address line of the address to validate. For example, "123 Main Street".
Address2	String	The Address2 of the address to validate. This will only be used in particular situations where the Address1 does not help in validating the address.
City	String	The city of the address to validate. For example, "New York". The city isn't required, but if one is not provided, the Zip code is required.
State	String	The state of the address to validate. For example, "NY". This does not need to be contracted, full state names will work as well. The state isn't required, but if one is not provided, the Zip code is required.
PostalCode	String	The zip code of the address to validate. A zip code isn't required, but if one is not provided, the City and State are required.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

ValidateAddressWithSuiteLink Outputs

Name	Type	Values	Description
Address	String	Varies	The corrected Address line 1.
City	String	Varies	The corrected city name.
State	String	Varies	The corrected state name.
Zip	String	Varies	The corrected zip code + 4.
Address2	String	Varies	The corrected Address line 2.
BarcodeDigits	String	Varies	The post office delivery barcode digits.

CarrierRoute	String	4 chars	4 chars: 1 for the route type, 3 for the route code. Identifies a group of addresses when prepended by 5-digit Zip.
CongressCode	String	Varies	The congress code of the given address.
CountyCode	String	Varies	The county code of the given address.
CountyName	String	Varies	The name of the county in which the given address lies.
Fragment	String	Varies	The parsed "Fragment" box, apartment or unit number. Same as FragmentPMBNumber.
FragmentHouse	String	Varies	The parsed house number of the given address.
FragmentPMBNumber	String	Varies	The parsed apartment, box, unit, etc. number of the given address.
FragmentPMBPrefix	String	Varies	The parsed type of the apartment, box, unit, etc. For example, "APT" or "BOX".
FragmentPostDir	String	Varies	The parsed post-directional of the address's street. "West" in "North Main St West".
FragmentPreDir	String	Varies	The parsed pre-directional of the address's street. "North" in "North Main St West".
FragmentStreet	String	Varies	The parsed name of the street in the given address. "Main" in "North Main St West".
FragmentSuffix	String	Varies	The parsed suffix of the street in the given address. "St" in "North Main St West".
DPV*	String	1-8	Number that correlates to a DPV result.
DPVDesc*	String	Varies	Explains DPV result.
DPVNotes*	String	1-14	Number that correlates to DPV notes description.
DPVNotesDesc*	String	Varies	Details about the DPV result.
Corrections*	String	1-43	Number that correlates to a Corrections Description.
CorrectionsDesc*	String	Varies	Description of what was corrected in an address.
Error – Desc	String	Varies	If there was an internal web service error, the description

			will be displayed here.
Error – Number	String	1, 2, 3, 4, 5	See “Error Codes” below.
Error – Location	String	Always null	Deprecated, no longer used.

*Possible values/descriptions listed below.

DPV Codes

DPV - DPV Desc:
<p>"1", "Yes, the input record is a valid mailing address"</p> <p>"2", "No, the input record is not in the DPV database of valid mailing addresses"</p> <p>"3", "The apartment or rural route box number is not valid, although the house number or rural route is valid"</p> <p>"4", "The input record is a valid mailing address, but is missing the apartment or rural route box number"</p> <p>"5", "Internal error. Please contact customer service"</p> <p>"6", "Internal error. Please contact customer service"</p> <p>"7", "Internal error. Please contact customer service"</p> <p>"8", "Internal error. Please contact customer service"</p>

DPVNotes - DPVNotesDesc:
<p>"1", "The input address matched the ZIP+4 record"</p> <p>"2", "The input address did not match the ZIP+4 record"</p> <p>"3", "The input address matched the DPV record"</p> <p>"4", "The input address primary matched DPV but the secondary did not"</p> <p>"5", "The input address is a military APO/FPO address"</p> <p>"6", "The input address is a general delivery address"</p> <p>"7", "The input address is a building name, and the primary is missing"</p> <p>"8", "The input address matched a DPV high-rise but a secondary was not input"</p> <p>"9", "The input address primary is invalid"</p> <p>"10", "The input address is missing a PO box, rural route, or highway contract box number"</p> <p>"11", "The input address (PO box, rural route, or highway contract) primary is invalid"</p> <p>"12", "The input address specified a PMB and matched a CMRA"</p> <p>"13", "The input address did not include a PMB and matched a CMRA"</p> <p>"14", "The input address matched and retained a unique ZIP code"</p> <p>"15", "SuiteLink did not find Suite or Unit data to append to the address"</p>

Corrections - CorrectionsDesc:

"1", "State not found"
"2", "City not found"
"3", "Street not found"
"4", "Address not found"
"5", "Uncomputable +4 range"
"6", "+4 unavailable"
"7", "Multiple streets match"
"8", "Multiple addresses match"
"9", "Time ran out"
"10", "Output too long"
"11", "Unused and reserved"
"12", "Company phonetic match used"
"13", "First company match used"
"14", "State determined from city"
"15", "State determined from ZIP"
"16", "City phonetic match used"
"17", "City determined from ZIP"
"18", "Acceptable city name used"
"19", "Street phonetic match used"
"20", "Predirection dropped"
"21", "Predirection added"
"22", "Suffix dropped"
"23", "Suffix added"
"24", "Postdirection dropped"
"25", "Postdirection added"
"26", "Predirection standardized"
"27", "Street standardized"
"28", "Suffix standardized"
"29", "Postdirection standardized"
"30", "Street missing"
"31", "Alternate address used"
"32", "Unit standardized"
"33", "Unit missing"
"34", "Unit not found"
"35", "Unit unverified"
"36", "Rural box not found"
"37", "Leftovers found"
"38", "ZIP not found"
"39", "ZIP changed"
"40", "+4 changed"
"41", "Route changed"
"42", "Suite or Unit data appended using SuiteLink"
"43", "Directional or Suffix is either missing or unused"

Error Codes

Error codes in AV2 are the same for all operations. They are as follows:

Error Code 1 – “Input cannot be less than zero length”

This error means the web service did not get any input. The connection to the service was made, and data was transferred, but no parameters were passed that the service could understand.

This error often happens when input is passed to the service with namespaces that the service does not understand. Applying a namespace to any of the parameters (Address, City, State, etc.) will cause this error. Additionally, requests made in the “rpc/encoded” format will cause this error. The only namespace that should appear in any element is the “<http://www.serviceobjects.com>” namespace on the root Validate* element as so:

```
<ValidateAddress xmlns="http://www.serviceobjects.com/">
```

Note, however, that the namespace is not applied to the ValidateAddress element, it is only present.

Error Code 2 – Various descriptions

This error code appears when various errors occur, but are of the expected nature. Oftentimes, malformed or incomplete input will cause an error 2.

The following is a list of the possible Error Descriptions that may accompany an Error Code 2:

Bad or Incomplete Addresses Errors:

“Address and Address2 fields were too long. Together, they must be 100 characters or less.”

“Address field was too long, must be 100 characters or less.”

“Address not found”

“Firm/Highrise name conflict”

“Multiple addresses match”

“Insufficient address data”

“Please enter a valid address number.”

“Please input a street address.”

“Please input either zip code or both city and state.”

“PR Urb not found in city street list”

“Street not found”

“Street Not Found”

“Street number or box number out of range”

Failed Authentication Errors:

“Please provide a valid license key for this web service.”

“The daily allowable number of transactions for this license key has been exceeded.”

“The monthly allowable number of transactions for this license key has been exceeded.”

“The total allowable number of transactions for this license key has been exceeded.”

“There are not enough transactions available. Check your daily/monthly transaction limits.”

“This license key has expired.”

“This license key has not yet been activated.”

“This operation is limited to specific usage. Please contact customer service to have your key enabled.”

“Your license key does not work on this service.”

Web Service Invocation Errors:

“Address could not be found, search timed out.”

“City has no streets (Unable to load street list)”

“Could not parse address line. Internal error “

“Error initializing service”

“Internal error.”

“Server busy. Please try your request again.”

“This operation is not supported with this data provider.”

“Multiple addresses match” means that there was an ambiguity about the address, most likely having to do with the street directional. Often, if there is a “West Main St” and an “East Main St” in a given location, trying to validate just “Main St” will give a “multiple addresses match” error.

Error Code 3/4/5 – Various descriptions

An error code 3, 4, or 5 is a fatal error and it means something serious has gone wrong. You will never see one of these error codes in a live production environment.

Integration

Integrating AV into your application should be easy and straightforward. If you are using a common platform, ServiceObjects may already have sample code built that you can use:

http://www.serviceobjects.com/support/dots_example_code.asp

However, if you are using a common platform that does not already have sample code, you can ask ServiceObjects to build you an example. Email support@serviceobjects.com for more details.

Frequently Asked Questions

Which Operation Should I Use?

Picking which operation you want to use should be decided carefully. Depending on your environment and needs, you will need to use different operations for their corresponding strengths. As of this writing, we recommend `ValidateAddressWithDPV` as it provides standardization, validation, fragment returns, DPV descriptions, and corrections.

If you only want to standardize your address and don't mind if it's not “DPV” valid, you could use our “`ParseAddress`” operation. If your addresses are not parsed

out into address1, address2, city, state, and zip code, you could use our ValidateAddressSingleLine operation and put the entire address string into one line.

If you have an Address2 field, you can use any operation with the proper input. See the question below for more information.

I Have Address2 elements. What should I do? How do I give you Address2 if I'm using an operation that doesn't have an Address2 input?

You can either switch operations, or you can concatenate your Address2 data onto your Address1 line, separated by a comma. The operation will interpret everything after the comma as secondary address information.

If you need fragment elements, it is highly recommended that you use ValidateAddressWithDPV or ValidateAddressWithFragments. If not, use ValidateAddressWithExtras to save you from having to concatenate the Address2 to the Address1 field.

For example, if your address is:

123 Main Street
Building C
Anytown, CA 99999

You could pass it in as:

123 Main Street, Building C
Anytown
CA
99999

I Have an Address3. What should I do with it?

We currently don't support any handling of the Address3 field. Passing it in as part of either the Address1 or Address2 fields is not recommended, as it may inhibit validation. If this information must be included, then your best option is to append all your address lines together separated by spaces, and sent through our ValidateAddressSingleLine operation along with city, state, and zip code appended.

The Sample Code is Giving Strange Errors or is Crashing!

Most likely, the sample code cannot connect to ServiceObjects. Many environments will not allow you to connect out on port 80, or will clip out XML data from these requests/responses.

The easiest way to check for this is to open a browser on the machine running the sample code. In your browser, navigate to:

<http://trial.serviceobjects.com/av2/AddressValidate.aspx>

Then try to run one of the operations with your trial key. If you get a browser error, or get no data back, then the sample code isn't able to connect either. Contact your systems administrator to resolve why you are not able to connect to ServiceObjects.

Address Validation says it can't find my street!

DOTS Address Validation doesn't know about every address, especially empty lots or new streets. Often, it won't be able to validate these locations. In general, we are as good as the USPS at identifying addresses. If they have an address we don't, we will have it within 30 days after our monthly update is run.

Does Address Validation do delivery point validation? I need to know if the USPS can deliver to this address.

Yes, we do. We recommend using our "ValidateAddressWithDPV" operation, but our "ValidateAddressWithSuiteLink" and "ValidateAddressWithRDI" operations also include DPV information.

Is Address Validation CASS-certified? Can I use it to get postal discounts?

Yes and No, our core address validation engine is CASS certified. It meets and exceeds all requirements for address correction except bulk mail discounts. AV does not print the UPSP forms required for bulk mail discounts, because of this we are not formally CASS certified. You cannot use AV for bulk-rate postal discounts.

What are the possible errors that AV will return if an address is invalid?

Please refer to "Error Codes", above.

What does "Multiple Addresses Match" mean? How do I get a single result?

“Multiple Addresses Match” means that AV found the address, but couldn’t resolve it from other addresses that were very similar, typically because of directional elements.

For example, if you validate “123 Main street, Anytown, CA”, but in Anytown, CA, there is only a “West Main street” and an “East Main street”, then this will result in a “Multiple Addresses Match”. You will need to specify either West Main or East Main as the street name to get a corrected, single address.

There are other cases in which “multiple addresses match” will occur, but are rare and often have to do with a difference in zip+4, or suite numbers.

In this case you can try calling our GetPossibleMatches operation to see the results, but if none of the results are DPV valid, GetPossibleMatches won’t return any results.

I need to know exactly how long each of the output fields could be. What is your standard field length?

Please email us at support@serviceobjects.com, and we’ll send you the exact field length specifications. As a general rule, you won’t get a field longer than 80 characters.

Does AV validate suite or apartment numbers?

If you use any of our DPV-enabled operations (ValidateAddressWithDPV, ValidateAddressWithRDI, ValidateAddressWithSuiteLink) then yes.

Does AV validate PO Boxes/box numbers?

Yes. PO boxes are correctable and are indeed validated. Out of range PO Box numbers will yield the appropriate error

I need both address correction and geocoding information. How can I get both at once?

We offer alternative services to provide this information. Please see “DOTS Address Validation Plus US” at this URL:

<http://www.serviceobjects.com/products/address/address-validation-plus-us>

Additionally, if we run batches for you (see question below) we can provide both address correction and geocoding simultaneously.

If you have a need for special information or format, email us at support@serviceobjects.com

Where do you get your address information?

We gather our data from many various sources. The short answer is the US Postal Service.

How often do you update your address data?

We'll update our information monthly, but in some special situations, we may update it more frequently.

I'm not a programmer. How do I use DOTS Address Validation US?

ServiceObjects runs batches for you! A free batch trial is available at <http://www.serviceobjects.com/batch/upload>.

Conclusion

ServiceObjects is proud to offer you a free trial of DOTS Address Validation 2 - US.

Sign up today for a free trial at:

<http://www.serviceobjects.com/products/address/address-validation-us>

Other technical questions or concerns can be directed to support@serviceobjects.com.

If you are interested in purchasing DOTS Address Validation 2 US, please contact sales@serviceobjects.com.

We welcome your feedback! Please do not hesitate to let us know what you think of our web services, documentation, or customer support.

Service Objects, Inc.
Insight on Demand

133 E. De la Guerra St., #10
Santa Barbara, CA 93101
Tel: 805.963.1700
Fax: 805.963.9179
www.serviceobjects.com