MAILROOM TOOLKIT®

Integrated Address Management Solutions

Web Service Guide

United States

Satori Software®

ZIPService

The ZIPService is a Web Service interface to the ZIPTask library. Satori Software provides this interface as an alternative to the COM and .NET interfaces. This service takes a single address, corrects it using a CASS Certified address correction process and returns it to you.

The ZIPService uses a single function: CheckAddress. This function takes an address, your registration and Add-on key credentials and a properties object. It returns the processed address, along with ZIP + 4, carrier route and barcode information. With an add-on, you can retrieve geocode information as well. With just a few lines of code, you can process an address to make it more complete and more likely to arrive at its destination.

Overview

Use the following procedure to process an address with the ZIPService:

1. Add a Web reference to the ZIPService. The address of the service is https://ws.satorisoftware.com/Architect/us/ZIP/ZIPService.asmx.
2. Create a ZIPService object.
3. Create a ZIPService.Credentials object and assign the ProductKey and AddOnKeys.
4. Create a ZIPService.ZIPServiceProperties object and set the properties as you wish. We recommend that you set all properties. See the Properties section for more information.
5. Create a ZIPService.ZIPAddress object and build your address with this object.
6. Call CheckAddress on these three objects. Make sure you store the return value using a ZIPAddressOutput object.

ZIPService Functions

Below are the methods, properties and fields available in ZIPService. If you have added a Web reference to the ZIPService in Visual Studio, you can view all of these functions in the Object Browser.

Note that in the function below, we use the namespace MyZipServiceReference. This may be different on your system, depending on the development environment. The function example below assumes that the development environment is C#; other environments may use different syntax.

CheckAddress

Syntax

```
MyZipServiceReference.ZipAddressOutput
CheckAddress(MyZipServiceReference.Credentials,
MyZipServiceReference.ZIPServiceProperties,
MyZipServiceReference.ZIPAddress);
```

Description

Processes a single address and attempts to correct it.

Parameters

- Credentials – A MyZipServiceReference.Credentials object that contains two objects: ProductKey, a string that contains your MailRoom ToolKit Architect license key, and AddOnKeys, an array of strings that contain license keys for your Add-ons.
- ZIPServiceProperties – A MyZipServiceReference.ZIPServiceProperties object that contains the settings for all properties. We recommend that you set all property values. See below for the list of available properties.
- ZIPAddress – A MyZipServiceReference.ZIPAddress object that contains the address you want to process. See below for the available fields in this object.

Return values

A MyZipServiceReference.ZIPAddressOutput object that contains your processed address. See below for the valid fields in this object.

Notes

See also

Properties

The properties listed below are values within the ZIPServiceProperties structure. To set properties for your ZIPService process, instantiate a ZIPServiceProperties object, set the member properties listed below and pass this structure to the CheckAddress() function.

We recommend that you set all property values. Any property that you do not set will contain the default value, which is usually false, 0 or the first enumeration value. However, this may not always be the case for all programming environments. So, to make sure that you receive the best results for your application, you should set every property.

Note that in all the properties below, we use the namespace MyZipServiceReference. This may be different on your system, depending on the development environment.

AddressInputPreference

Data type

DualAddressValue (an enumeration)

Description

Determines whether to prefer the street address or PO Box for addresses that have both.

Notes

Use one of the following to set this property:

- AddressInputPreference.eByPosition – The bottom address will be used, regardless of which type it is.
- AddressInputPreference.ePreferPOBox – Uses PO Box addresses.
- AddressInputPreference.ePreferStreet – Uses street addresses.

ApplyCasingBusiness

Data type

boolean

Description

Determines whether to apply the Casing property to the Business field.

Notes

AssignLOT

Data type

boolean

Description

Determines whether to assign the Line-of-Travel number to processed addresses.

Notes

AssignRDI

Data type

boolean

Description

Determines whether to look up the RDI for this address, which indicates if the address is residential or commercial.

Notes

- This property requires an Add-on.
- You must purchase a separate data file from the USPS to retrieve this information.

Casing

Data type

CasingValue

Description

Determines the case format for the returned address.

Notes

Use one of the following:

- MyZipServiceReference.CasingValue.eUpper – The address will be returned in all UPPER CASE.
- MyZipServiceReference.CasingValue.eLower – The address will be returned in all lower case.

- MyZipServiceReference.CasingValue.eMixed – The address will be returned in Mixed Case.

CityFormat

Data type

CityFormatValue

Description

Determines the format of the returned City field.

Notes

Use one of the following:

- MyZipServiceReference.CityFormatValue.eFullName – Returns the full city name.
- MyZipServiceReference.CityFormatValue.eAbbreviated – Returns the abbreviated name if the city name contains more than 13 characters.
- MyZipServiceReference.CityFormatValue.eAsInput – THIS VALUE IS DEPRECATED.

DPVFailureAsError

Data type

boolean

Description

Determines whether to treat addresses whose secondary address data fails DPV processing as errors.

Notes

- The default for this property is false.
- Addresses with missing or invalid secondary information can still be successfully corrected and assigned a ZIP + 4 Code.

FirmOutput

Data type

FirmOutputValue

Description

Determines how firm information is returned for a corrected address.

Notes

Use one of the following:

- MyZipServiceReference.FirmOutputValue.eAsInput – Returns the business name sent as input.
- MyZipServiceReference.FirmOutputValue.eBusinessNameIfBlank – Returns firm information only if the business field was blank on input.
- MyZipServiceReference.FirmOutputValue.eBusinessNameAlways – Overwrites the contents of the business field with any firm information found.

HighwayContractFormat

Data type

ElementFormatValue

Description

Determines how the service will format highway contract information.

Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

KeepAliasAddress

Data type

boolean

Description

Determines whether an input address with a valid street name alias will be allowed or whether they will be replaced with the official USPS street name.

Notes

KeepExtraPrimaryData

Data type

boolean

Description

Determines whether to keep any additional and possibly extraneous information in the primary address line.

Notes

KeepNonMailingCity

Data type

boolean

Description

Determines if valid but not preferred city names will be returned for corrected addresses.

Notes

PMBOutput

Data type

PMBOutputValue

Description

Determines the output location of personal mailbox information, if any.

Notes

Use one of the following:

- MyZipServiceReference.PMBOutputValue.eWithUnit – PMB information outputs on the same line as the unit information.
- MyZipServiceReference.PMBOutputValue.eAsInput – PMB information outputs the same as in the input address.

POBoxFormat

Data type

ElementFormatValue

Description

Determines how the service will output PO Box information.

Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

PostDirectionalFormat

Data type

ElementFormatValue

Description

Determines how the service will format post-directional information.

Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

PreDirectionalFormat

Data type

ElementFormatValue

Description

Determines how the service will format pre-directional information.

Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

PrimaryAddressOutput

Data type

PrimaryAddressOutputValue

Description

Determines how the service will output the primary address information.

Notes

Set this to one of the following:

- MyZipServiceReference.PrimaryAddressOutputValue.eAddressLine1 – Returns the primary address information as the Address Line 1 field.
- MyZipServiceReference.PrimaryAddressOutputValue.eAddressLine2 – Returns the primary address information as the Address Line 2 field.

RuralRouteFormat

Data type

ElementFormatValue

Description

Determines how the service will format rural route address information.

Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.

- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

SuffixFormat

Data type

ElementFormatValue

Description

Determines how the service will format street suffix information.

Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

UnitDesignatorFormat

Data type

ElementFormatValue

Description

Determines how the service will format unit type information, such as Suite or Apartment.

Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

UpdateUncorrectedCityStZip

Data type

boolean

Description

Determines if updated City, State and ZIP Code information should be returned for addresses that could not be completely corrected.

Notes

UseDPV

Data type

boolean

Description

Determines if addresses should be verified by the DPV process.

Notes

- DPV processing is required for an address to be corrected and assigned a ZIP + 4 Code.
- We recommend that you set this property to true.
- Without DPV processing, an address will only be confirmed as within a valid range of addresses. DPV processing confirms that the specific street address is a valid delivery point.

UseGeocode

Data type

boolean

Description

Determines if the service will return longitude and latitude information for the corrected address.

Notes

Geocode requires an add-on from Satori Software.

UseLACS

Data type

boolean

Description

Determines if the service will look up LACS^{Link} information.

Notes

- LACS^{Link} processing is required for an address to be corrected and assigned a ZIP + 4 Code.
- We recommend that you set this property to true.

UseSuiteLink

Data type

boolean

Description

Determines if the service should use Suite^{Link} to locate missing secondary information based on the business field.

Notes

Fields

The items below are members of the ZIPAddress and ZIPAddressOutput structures. These items will be divided into two sections, one for each class. The ZIPAddressOutput structure contains all of the fields from the ZIPAddress structure, plus several additional fields.

Assign input address elements to the member objects in a ZIPAddress object before passing that object to the CheckAddress function. The CheckAddress function will return a ZIPAddressOutput structure. This object contains the corrected ZIPAddress fields as well as several other output-only fields.

For an effective ZIPService process, we recommend that you use at least the following fields as input: AddressLine1, City, State and ZipCode.

Note that in all the fields below, we use the namespace MyZipServiceReference. This may be different on your system, depending on the development environment.

ZIPAddress

BusinessName

Data type

string

Description

The name of the business at this address.

Notes

This will be used to fill in missing secondary information using Suite^{Link} if the useSuitelink property is set to True.

AddressLine1

Data type

string

Description

The first address line.

Notes

We recommend that you set this field to run the ZIPService.

AddressLine2

Data type

string

Description

The second address line.

Notes

This field is not required.

AddressBlock

Data type

string

Description

A complete address block, as would be printed on a mail piece.

Notes

- You can set this field instead of using the AddressLine1, City, State and ZipCode fields.
- Each line must be delimited by a carriage return or line feed character.

City

Data type

string

Description

The city of this address.

Notes

We recommend that you set this field to run the ZIPService.

State

Data type

string

Description

The state for this address.

Notes

We recommend that you set this field to run the ZIPService.

ZipCode

Data type

string

Description

The ZIP Code of this address.

Notes

We recommend that you set this field to run the ZIPService.

CityStateZip

Data type

string

Description

The City, State and ZIP Code fields formatted as a single line.

Notes

This is the same information as would print at the bottom of an address block on a mail piece.

CarrierRoute

Data type

string

Description

The carrier route information for this address.

Notes

Street

Data type

string

Description

The street name for this address.

Notes

This information will be created from the primary address information.

Suffix

Data type

string

Description

The street suffix (e.g. St., Ave., Ln.) for this address.

Notes

This information will be created from the primary address information.

UnitDesignator

Data type

string

Description

The type of unit (e.g. Apt., Ste., Fl.) for the secondary address information.

Notes

This information will be created from the secondary address information, if any.

UnitNumber

Data type

string

Description

The unit number for this address.

Notes

This information will be created from the secondary address information, if any.

Urbanization

Data type

string

Description

The urbanization for this address.

Notes

This field applies only to addresses in Puerto Rico.

ZIPAddressOutput

CarrtCoded

Data type

boolean

Description

Indicates whether this address was assigned carrier route information.

Notes

CensusBlock

Data type

string

Description

The name of census block in which this address is located.

Notes

- A census block is the smallest area for which the census bureau produces information.
- You must have the Geocode add-on installed to generate this information.

CensusTract

Data type

string

Description

The census tract in which an address is located.

Notes

- A census tract is a geographic area, often coinciding with the limits of cities or towns, defined for census purposes. A census tract can contain several census blocks.
- You must have the Geocode add-on installed to generate this information.

CongressionalDistrict

Data type

string

Description

The congressional district in which this address is located.

Notes

CountyCode

Data type

int

Description

The county code for this address.

Notes

CountyFIPSCode

Data type

string

Description

The FIPS (Federal Information Processing Standard) code for the county in which this address is located.

Notes

This is a 5-digit number. The first two digits represent the state, while the last three digits represent the county.

CountyName

Data type

string

Description

The name of the county in which this address is located.

Notes

DeliveryPointCheckDigit

Data type

string

Description

Returns the two-digit delivery point and one-digit check digit for the ZIP + 4 Code.

Notes

- The delivery point digits indicate a mail delivery point within a ZIP + 4 Code as to provide a unique number for every deliverable address.
- This information is used in POSTNET and Intelligent Mail barcodes.
- The check digit verifies that the ZIP + 4 Code is valid and does not contain an error.

DPBarcodeString

Data type

string

Description

Contains the delivery point barcode as a string.

Notes

- To print this data as a barcode, install the SatBar.ttf from the MailRoom ToolKit Architect disc and apply it to this field in 16 point font.
- This is also known as the POSTNET barcode.

DPVCoded

Data type

boolean

Description

Indicates that this address was confirmed as a valid delivery point.

Notes

DPVFootnotes

Data type

string

Description

Indicates the results of the DPV verification process.

Notes

Returns one or more of the following:

- LK – Processing locked out due to a seed record being processed
- AA – Matched to the ZIP+4 file
- A1 – No match against the ZIP+4 file
- BB – Matched to DPV file (all components confirmed)
- CC – Matched only after removing secondary Information; they were presented but invalid
- N1 – Input Primary matched, but high-rise missing secondary number
- M1 – Primary number missing
- M3 – Primary number invalid
- P1 – Input missing PO, RR, HC box number
- P3 – Failed DPV because of invalid PO, RR, or HC box number
- RR – Matched CMRA (found in CMRA file)
- R1 – Matched CMRA, but secondary number (i.e., PMB) missing
- U1 – Matched unique zip code
- G1 – Matched General delivery
- F1 – Matched military address

DPVIndicator

Data type

string

Description

Contains a single character that describes the result of DPV processing.

Notes

Returns one of the following:

- Y – Both the primary and secondary address data validated against the DPV database.
- S – The primary address is valid, but the secondary is invalid.
- D – The primary address is valid, but the address is missing secondary information.

- N – The primary address is not valid. This address was not given a ZIP + 4 Code.
- “” – The address was not presented to the DPV table because it was missing components needed for the lookup.
- X – The DPV database has been locked-out because of a protocol violation. You must unlock DPV before any more addresses can be processed with DPV.
- E – The DPV data file is more than 105 days old. By USPS restrictions, no more addresses can be presented to the DPV table

DPVIsCMRA

Data type

boolean

Description

Indicates whether this address is a commercial mail-receiving agent.

Notes

DPVIsNoStat

Data type

boolean

Description

If True, indicates that the address is in the database, but is not receiving mail yet.

Notes

This often indicates the address is new construction, currently being built but not yet occupied and receiving mail.

DPVIsVacant

Data type

boolean

Description

Indicates whether this address has been vacant for at least 90 days.

Notes

ErrorCode

Data type

int

Description

A numerical code that indicates the results of CASS processing.

Notes

See the appendix for the codes.

ErrorCodeString

Data type

string

Description

Text that explains the results from CASS processing.

Notes

This text corresponds to the ErrorCode field.

ExtraInfo

Data type

string

Description

Contains extra address information that the service was unable to parse into any other fields.

Notes

GeocodeFootnote

Data type

string

Description

Indicates the results of Geocode processing.

Notes

Requires the Geocode add-on.
Contains one of the following:

- 00 – Unable to perform a geocode lookup
- 03 – Geocode data based on a 3-digit ZIP Code
- 05 – Geocode data based on a 5-digit ZIP Code
- 07 – Geocode data based on a 7-digit ZIP Code
- 09 – Geocode data based on a 9-digit ZIP Code

IsResidence

Data type

boolean

Description

Indicates whether this address is residential or not.

Notes

To use this property, you must have the RDI add-on and subscribe to the RDI service from the USPS.

LACS

Data type

boolean

Description

Indicates whether this address was flagged by LACS^{Link} processing.

Notes

LACSFootnote

Data type

string

Description

Provides additional information about the LACS process.

Notes

Contains one of the following:

- blank – Not processed / Seed record.
- 00 – No match.
- 09 – Matched to default high-rise address; address not updated.
- 14 – Match failed to build new address.
- 92 – Match secondary dropped from input.
- A – Match success.

LACSIndicator

Data type

string

Description

Indicates the results of a LACS process.

Notes

Returns one of the following:

- blank – Not processed.
- N – Match, but there was a failure to build new address.
- Y – Match success, new address provided.
- S – Match with secondary dropped from input.
- F – Seed record.

Latitude

Data type

string

Description

The latitude of this address.

Notes

You must have the Geocode add-on installed to generate this information.

Longitude

Data type

string

Description

The longitude of this address.

Notes

You must have the Geocode add-on installed to generate this information.

LOTNumber

Data type

string

Description

The Line-of-Travel number for this address.

Notes

- Carrier route sorts require this information.
- To generate this data, you must have set the assignLOT property to True.

MatchedToDefault

Data type

boolean

Description

Indicates if this address matched to a default ZIP Code.

Notes

Default ZIP Codes are for large, single addresses, such as universities or corporations, that do not have +4 codes.

MSACode

Data type

string

Description

Contains the Metropolitan Statistical Area code for this address.

Notes

You must have the Geocode add-on installed to generate this information.

Plus4Coded

Data type

boolean

Description

Indicates whether this address received a valid ZIP + 4 Code.

Notes

PMB

Data type

string

Description

The private mail box number for this address, if any.

Notes

The default value for this property is an empty string.

PostDirectional

Data type

string

Description

Contains the post-directional indicator of this address, if any.

Notes

PreDirectional

Data type

string

Description

Contains the pre-directional indicator of this address, if any.

Notes

RecordType

Data type

string

Description

Indicates the type of address that this is.

Notes

Contains one of the following characters:

- S – Street record
- P – Post office box
- R – Rural Route or Highway Contract
- H – High-rise, Building or Apartment
- F – Firm Record
- G – General Delivery
- M – Multi-Carrier Record

SuiteLinkFootnote

Data type

string

Description

Indicates the results of SuiteLink matching.

Notes

- To generate this data, you must set the useSuitelink property to True.
- Contains one of the following:
  - blank – Was not processed by the SuiteLink engine: the address did not qualify for a lookup. Only default high rise addresses qualify for a SuiteLink lookup.
  - A – The address was processed and secondary information was added to the resulting address.
  - 00 – The address was processed through the SuiteLink engine, but did not result in a successful match; no secondary information was added.

CASSDate

Data type

int

Description

Encodes the date that this record was last processed.

Notes

CASSService

The CASSService is a Web Service interface to the CASSTask library. Satori Software provides this interface as an alternative to the COM and .NET interfaces. This service takes a group of addresses, processes them using a CASS Certified address correction process and returns the updated addresses to you.

With the CASSService, you can process an entire mailing list and prepare it for automation and presorted discounts. All addresses, if validated, will be returned with the correct address information, as well as accurate ZIP + 4 Codes and POSTNET barcodes. When the CASS validation completes, you can then retrieve an address correction report that the USPS will accept as valid documentation of address correction. You can configure the information returned using the SetProperty function. With the Geocode add-on, you can return latitude and longitude information as well.

With just a few lines of code, you can process a list of addresses to make them more complete, more likely to arrive at their destinations and eligible for discounts.

Overview

Use the following procedure to process an address with the CASSService:

1. Add a Web reference to the CASSService. The address of the service is http://ws.satorisoftware.com/Architect/US/CASS/CASSService.asmx.
2. Create a CASSService object.
3. Create a CASSService.Credentials object and assign the ProductKey and AddOnKeys.
4. Call CASSService.PrepareTask(MyCredentialsObj) to verify your registration keys.
5. Create a CASSService.CASSServiceProperties object and set the properties as you wish. We recommend that you set all properties. See the Properties section for more information.
6. Call CASSService.SetProperties(MyPropertiesObj) to pass your properties to the server.
7. Call CASSService.ValidateProperties() to verify that all properties are correct.
8. Create a CASSService. AddressBlock object and build your addresses with this object. The addresses in this object should all have the fields specified by the SETTINGS_FIELD_LIST_IN property. The block must have no more than SETTINGS_INPUT_BLOCK_RECORD_COUNT number of records. Create as many AddressBlock objects as you need. The records in all address blocks must total SETTINGS_RECORD_COUNT.
9. Call CASSService.Update on each AddressBlock object. This function will return these addresses as a block, with each record having the fields specified by the SETTINGS_FIELD_LIST_OUT property.
10. Save the address correction report with CASSService.GetReportsAsPDF.
11. Call CASSService.EndTask to free up all resources used on the server.

CASSService Functions

Below are the methods available in CASSService. If you have added a Web reference to the CASSService in Visual Studio, you can view all of these functions in the Object Browser.

Note that in the function below, we use the namespace MyCASSServiceReference. This may be different on your system, depending on the development environment. The function example below assumes that the development environment is C#; other environments may use different syntax.

EndTask

Syntax

CASSService.EndTask();

Description

Cleans up any server resources and removes all user data from the server.

Parameters

None.

Return values

None.

Notes

- EndTask will remove all of your address data from our servers to protect your privacy.
- You must call EndTask after you have called Update on your address blocks and retrieved the address correction report PDF.

See also

GetProperties

Syntax

PropertyValues MyCASSServiceReference.GetProperties(PropertyIDs);

Description

Retrieves the values of the specified properties.

Parameters

PropertyIDs – A list of the properties for which you want to retrieve the values.

Return values

GetPropertyValues – A list of the values of the properties specified in PropertyIDs.

Notes

See also

See the CASSService Properties section for the names and purposes of the CASSService properties.

GetReportsAsPDF

Syntax

PDFFile MyCASSServiceReference.GetReportsAsPDF();

Description

Retrieves the address correction report as a formatted PDF file.

Parameters

None.

Return values

PDFFile – A binary stream that you can save as a PDF file.

Notes

The CASS Summary Report (PS Form 3553) is the only CASS report currently available.

See also

PrepareTask

Syntax

MyCASSServiceReference.PrepareTask(
MyCASSServiceReference.Credentials);

Description

Validates your product and add-on keys and readies the server to accept additional function calls from your client application.

Parameters

Credentials – A MyCASSServiceReference.Credentials object that contains two objects: ProductKey, a string that contains your MailRoom ToolKit Architect license key, and AddOnKeys, an array of strings that contain license keys for your Add-ons.

Return values

None

Notes

- PrepareTask should be called only once, after you create the CASSService object.
- You must call this function before calling any of the other functions or setting any of the properties of CASSService. Failing to do so will cause subsequent function calls to fail.

See also

SetProperties

Syntax

MyCASSServiceReference.SetProperties(Properties);

Description

Sets the values of the specified properties.

Parameters

Properties – An array of property and value pairs.

Return values

None.

Notes

We recommend that you set all properties.

See also

See the CASSService Properties section for the enum names and purposes of the CASSService properties.

Update

Syntax

CorrectedAddressBlock
MyCASSServiceReference.Update(UncorrectedAddressBlock);

Description

Corrects the addresses contained in the AddressBlock object.

Parameters

UncorrectedAddressBlock – A block of uncorrected addresses. Each address should have the fields listed in SETTINGS_FIELD_LIST_IN.

Return values

CorrectedAddressBlock – The input addresses, corrected and updated. Each address will have the fields specified in SETTINGS_FIELD_LIST_OUT.

Notes

- Call this function after ValidateProperties. You should call this function for each block of addresses in your mailing list.
- You may want to experiment with the SETTINGS_INPUT_BLOCK_RECORD_COUNT property. In tests, we have found the optimal setting to be around 25-50. The number of fields that you want returned (see SETTINGS_FIELD_LIST_OUT property) greatly affects this number.
- For optimal performance, only ask for the output fields that you need. Extra information requires additional lookups that slow processing.

See also

- See the CASSAssembly Properties section for the definition of:
    - SETTINGS_FIELD_LIST_IN
    - SETTINGS_FIELD_LIST_OUT
    - SETTINGS_INPUT_BLOCK_RECORD_COUNT

ValidateProperties

Syntax

Validated MyCASSServiceReference.ValidateProperties();

Description

Validates that the settings will work as configured.

Parameters

None.

Return values

Nothing.

Notes

- This function verifies that the basic requirements of a CASSService object have been met:
    - The address matching engine is loaded and able to run.
    - The input field list consists of the minimum set of fields, namely, ADDRESS_LINE_1 and either CITY and STATE or ZIP_CODE or LAST_LINE.
- This function needs to be called after you call SetProperties, but before you call Update.

See also

- SetProperty function
- See the CASSService Properties section for the enum names and purposes of the CASSService properties.

CASSService Properties

The properties listed below are values within the CASSServiceProperties structure. To set properties for your CASSService process, instantiate a CASSServiceProperties object, set the member properties listed below and pass this structure to the SetProperties() function.

We recommend that you set all property values. Any property that you do not set will contain the default value, which is usually false, 0 or the first enumeration value. However, this may not always be the case for all programming environments. So, to make sure that you receive the best results for your application, you should set every property.

Note that in all the properties below, we use the namespace MyCASSServiceReference. This may be different on your system, depending on the development environment.

CASS_ABBREVIATE_ADDRESS_LINE

Data type

boolean

Description

Determines whether to abbreviate the address line to 30 characters.

Notes

- Address lines of less than 30 characters will not be abbreviated.
- This property affects the fields ADDRESS_LINE_1, ADDRESS_LINE_2 and ADDRESS_BLOCK.

CASS_CERTIFY_FLAG

Data type

CertifyFlagOption (an enumeration)

Description

Determines if previously corrected addresses should be processed.

Notes

- You must include the CASSDATE field as part of the SETTINGS_FIELD_LIST_IN list.
- For large, regularly-processed lists, this property can improve processing speed significantly.
- Use one of the following values:

- CertifyFlagOption.eSkip – Check only those records not corrected with this issue.
- CertifyFlagOption.eCheckAllRecords – Check every record, regardless of date last corrected.
- CertifyFlagOption.eRebuild – Does not correct any records. Flags all records without ZIP + 4 Codes or corrected before the current issue as errors (error code set to 101). The CASSService then rebuilds the PS Form 3553 to only show as corrected those records that have been corrected with this issue.

CASS_DPV_FAILURE_AS_ERROR

Data type

boolean

Description

Determines whether to treat addresses whose secondary address (apartment, suite, etc.) fails DPV processing as errors.

Notes

Addresses that pass primary DPV processing but fail secondary checks can still be assigned a ZIP + 4 Code and qualify for presorted discounts.

CASS_DUAL_ADDRESS_INPUT_PREFERENCE

Data type

DualAddress (an enumeration)

Description

Determines whether to use the street address or PO Box as the primary address for those addresses that have both.

Notes

Use one of the following to set this property:

- DualAddress.eByPosition – The bottom address will be used, regardless of which type it is.
- DualAddress.ePreferPOBox – Uses PO Box addresses.
- DualAddress.ePreferStreet – Uses street addresses.

CASS_KEEP_ALIAS_ADDRESS

Data type

boolean

Description

Determines whether to keep the valid but unofficial street name alias given as input or to replace it with the official USPS street name.

Notes

CASS_KEEP_EXTRA_PRIMARY_DATA

Data type

boolean

Description

Determines whether to keep any additional and possibly extraneous information in the primary address line.

Notes

CASS_LIST_NAME

Data type

string

Description

The name of the mailing list to process.

Notes

This value will be printed on the PS Form 3553.

CASS_LIST_PROCESSOR

Data type

string

Description

The name of the person processing this list with address correction.

Notes

This value will be printed on the PS Form 3553.

CASS_MAILERS_ADDRESS

Data type

string

Description

The street address of the person or business sending mail to this list.

Notes

This value will be printed on the PS Form 3553.

CASS_MAILERS_CITY

Data type

string

Description

The city of the person or business sending mail to this list.

Notes

This value will be printed on the PS Form 3553.

CASS_MAILERS_NAME

Data type

string

Description

The name of the person or business sending mail to this list.

Notes

This value will be printed on the PS Form 3553.

CASS_MAILERS_STATE

Data type

string

Description

The state of the person or business sending mail to this list.

Notes

This value will be printed on the PS Form 3553.

CASS_MAILERS_ZIP

Data type

string

Description

The ZIP Code of the person or business sending mail to this list.

Notes

This value will be printed on the PS Form 3553.

CASS_UPDATE_UNCORRECTED_CITY_ST_ZIP

Data type

boolean

Description

Determines if city, state and ZIP Code information should be updated for addresses that could not be fully corrected and validated.

Notes

CASS_USE_SUITELINK

Data type

boolean

Description

Determines if the CASSService should use Suite^{Link} to look up missing secondary address information.

Notes

FORMAT_CASING

Data type

Capitalization (an enumeration)

Description

Determines the case in which addresses are returned.

Notes

Use one of the following:

- Capitalization.eCapUppper – Upper case.
- Capitalization.eCapLower – Lower case.
- Capitalization.eCapMixed – Mixed case.

FORMAT_CITY

Data type

AbbreviateCity (an enumeration)

Description

Determines whether to abbreviate the city name.

Notes

Use one of the following:

- AbbreviateCity.eCityAbbrNever – Always return the full city name.
- AbbreviateCity.eCityAbbrForce – Return the abbreviated city name if one exists.
- AbbreviateCity.eCityAbbrInput – THIS VALUE IS DEPRECATED.

FORMAT_FIRM_OUTPUT

Data type

FirmOutput (an enumeration)

Description

Determines where to ouput a firm name in a corrected record.

Notes

- This property applies to firm names found in corrected addresses, but not input as the BUSINESS field.
- Use one of the following:
  - FirmOutput.eFirmMoveToBusiness – Return as the BUSINESS field only if that field was input as blank.
  - FirmOutput.eFirmMoveToBusinessOverwrite – Always overwrite the BUSINESS field.

FORMAT_HIGHWAY_CONTRACT

Data type

AddressElementFormat (an enumeration)

Description

Determines the format of returned highway contract addresses.

Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and puctuation.
- AddressElementFormat.eFullWord – Always returns full words.

FORMAT_PMB_OUTPUT

Data type

PMBOutput (an enumeration)

Description

Determines the location of the private mailbox number.

Notes

Use one of the following:

- PMBOutput.ePMBWithUnits – Return on the same line as the unit information.

FORMAT_PO_BOX

Data type

AddressElementFormat (an enumeration)

Description

Determines how to format a PO Box address.

Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and puctuation.
- AddressElementFormat.eFullWord – Always returns full words.

FORMAT_POST_DIRECTIONAL

Data type

AddressElementFormat (an enumeration)

Description

Determines how to format the returned post-directional, if any.

Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and puctuation.
- AddressElementFormat.eFullWord – Always returns full words.

FORMAT_PRE_DIRECTIONAL

Data type

AddressElementFormat (an enumeration)

Description

Determines how to format the returned pre-directional, if any.

Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and puctuation.
- AddressElementFormat.eFullWord – Always returns full words.

FORMAT_PRIMARY_ADDRESS_OUTPUT

Data type

PrimaryAddOutput (an enumeration)

Description

Determines the location of the primary address information.

Notes

Use one of the following:

- PrimaryAddOutput.eTopJustified – Return as ADDRESS_LINE_1 if ADDRESS_LINE_2 is blank. Else, return primary address data as ADDRESS_LINE_2 and secondary address data as ADDRESS_LINE_1.
- PrimaryAddOutput.eAddressLine2 – Always return as ADDRESS_LINE_2.

FORMAT_RURAL_ROUTE

Data type

AddressElementFormat (an enumeration)

Description

Determines how to format returned rural route addresses.

Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and puctuation.
- AddressElementFormat.eFullWord – Always returns full words.

FORMAT_SUFFIX

Data type

AddressElementFormat (an enumeration)

Description

Determines how to format the returned street suffix.

Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and puctuation.
- AddressElementFormat.eFullWord – Always returns full words.

FORMAT_UNIT_DESIGNATOR

Data type

AddressElementFormat (an enumeration)

Description

Determines how to format the returned unit type.

Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and puctuation.
- AddressElementFormat.eFullWord – Always returns full words.

FORMAT_UNIT_OUTPUT

Data type

UnitOutput (an enumeration)

Description

Determines where to return the unit information for an address.

Notes

Use one of the following:

- UnitOutput.ePrimaryAddressLine – Return at the end of the primary address line.
- UnitOutput.eSecondaryAddressLine – Return as the secondary address line if that line is blank.

- UnitOutput.eSecondaryAddressLineOverwrite – Overwrite the secondary address line with the unit information.

FORMAT_UPDATE_CASE_BUSINESS

Data type

boolean

Description

Determines if the casing specified by FORMAT_CASING applies to the BUSINESS field.

Notes

FORMAT_UPDATE_CASE_NAMES

Data type

boolean

Description

Determines if the casing specified by FORMAT_CASING applies to the FIRST_NAME and LAST_NAME fields.

Notes

SETTINGS_FIELD_LIST_IN

Data type

ArrayOfField

Description

An array of fields that will be included in every input address record.

Notes

- At minimum, this array should include ADDRESS_LINE_1 plus either CITY and STATE, ZIP_CODE or LAST_LINE.
- See the Fields section of this document for a list of fields that can be included.

SETTINGS_FIELD_LIST_OUT

Data type

ArrayOfField

Description

An array of fields that will compose each returned address record.

Notes

See the Fields section of this document for a list of fields that can be included.

SETTINGS_INPUT_BLOCK_RECORD_COUNT

Data type

int

Description

Indicates the number of records that will be sent and returned with each Update call.

Notes

You must set this property.

SETTINGS_RECORD_COUNT

Data type

int

Description

Indicates the total number of records to be processed across all Update calls.

Notes

You must set this property.

Fields

The items below are part of the Field enumeration. Include these names in the ArrayOfFields passed to either SETTINGS_FIELD_LIST_IN or SETTINGS_FIELD_LIST_OUT to determine which fields CASSService looks to process from incoming addresses or returns in processed records. All field values are strings.

Many of these fields are output only. The descriptions below will indicate which fields do not accept input.

For an effective CASSService process, we recommend that you use at least the following fields as input: ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, STATE and ZIP_CODE.

Note that in all the fields below, we use the namespace CASSServiceReference. This may be different on your system, depending on the development environment.

The following fields must be in the list of input fields if you wish to include them in the list of output fields:

- RECORD_ID
- All USER_DEFINED fields
- COUNTRY
- NAME_SALUTATION
- MIDDLE_NAME
- NAME_SUFFIX

ADDRESS_BLOCK

Description

The full address as would be printed on the front of a mail piece.

Notes

You can pass or retrieve this value instead of ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, STATE and ZIP_CODE.

ADDRESS_LINE_1

Description

The first line of this address.

Notes

- We recommend that you set this field for each address.
- This will be the primary address information unless you have unit information on a separate line.

ADDRESS_LINE_2

Description

The second line of this address.

Notes

- We recommend that you set this field for each address.
- If you have unit information on a separate line, this may contain the street address.

BUSINESS

Description

The name of the business, if any.

Notes

Depending on the value of FORMAT_FIRM_OUTPUT, Suite^{Link} may return this information for a corrected address.

CARRIER_ROUTE

Description

The carrier route for this address.

Notes

- This will be returned for a successfully corrected address.
- Output only.

CASSDATE

Description

An encoded string that contains the date that this record was last processed.

Notes

Output only.

CITY

Description

The city of this address.

Notes

We recommend that you set this field for each address.

CITY_ABBREVIATED

Description

The city name, abbreviated to 30 characters.

Notes

Output only.

CONGRESSIONAL_DISTRICT

Description

The congressional district for this address.

Notes

Output only.

COUNTRY

Description

The country for this address.

Notes

For most addresses, this will be United States.

COUNTY_CODE

Description

The five-digit county code for this address.

Notes

Output only.

COUNTY_NAME

Description

The name of the county in which this address resides.

Notes

DPC

Description

The delivery point/check digit for this address.

Notes

Output only.

DPV_CODED

Description

Indicated if DPV processing confirmed the deliverability of this address.

Notes

Output only.

DPV_FOOTNOTE

Description

One or more two character strings that indicate the results of DPV processing.

Notes

- Output only.
- Returns one of the following:

  - LK – Processing locked out due to a seed record being processed.
  - AA – Matched to the ZIP+4 file
  - A1 – No match against the ZIP+4 file
  - BB – Matched to DPV file (all components confirmed)
  - CC – Matched only after removing secondary information; they were presented but invalid.
  - N1 – Input primary matched, but high-rise missing secondary number.
  - M1 – Primary number missing.
  - M3 – Primary number invalid.
  - P1 – Input missing PO, RR or HC box number.
  - P3 – Failed DPV because of invalid PO, RR or HC box number.
  - RR – Matched CMRA (found in CMRA file).
  - R1 – Matched CMRA, but secondary number (i.e., PMB) missing.
  - U1 – Matched unique ZIP Code.
  - G1 – Matched general delivery.
  - F1 – Matched military address.

DPV_INDICATOR

Description

Indicates how detailed the DPV match is.

Notes

- Output only.
- Returns one of the following:
  - Y – Both the primary and secondary (if present) validated against the DPV database.
  - S – The primary address is valid according to DPV, but the secondary is invalid.
  - D – The primary address is valid according to DPV, but the address is missing secondary information.
  - N – The primary address is not valid according to DPV.
  - “” – The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not ZIP+4 coded.
  - X – The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.
  - E – The DPV data file is more than 105 days old; by USPS restrictions, no more addresses can be presented to the DPV table.

DPV_IS_CMRA

Description

Indicates if the address is a commercial mail-receiving agent.

Notes

Output only.

DPV_IS_NOSTAT

Description

Indicates whether the address is a new construction that does not yet have regular mail delivery.

Notes

Output only.

DPV_IS_VACANT

Description

Indicates if this address has been unoccupied for at least 90 days.

Notes

Output only.

DP_BARCODE

Description

The delivery point barcode for this address.

Notes

- Output only.
- Apply the SATBAR font, size 12, to this string to print a POSTNET barcode.

ERROR_CODE

Description

An error code for this address.

Notes

- Every address will receive an error code, regardless of whether it was corrected successfully or not.
- Output only.
- See the Error Code table in the Appendix for a listing of all error codes and descriptions.

ERROR_STRING

Description

A description of the error code for this address.

Notes

- Every address will receive an error code, regardless of whether it was corrected successfully or not.
- Output only.
- See the Error Code table in the Appendix for a listing of all error codes and descriptions.

EWS_CODED

Description

Indicates that an address is on a new street not yet included in the USPS databases.

Notes

Output only.

EXTRA_INFO

Description

Contains any extra information from the input address that the CASSService was not able to parse into an address field.

Notes

Output only.

FIRST_NAME

Description

The first name of the resident of this address.

Notes

This field will not be processed and will be returned the same as input.

GEOCODE_CENSUS_BLOCK

Description

The census block for this address.

Notes

- A census block is the smallest area for which the census bureau produces information.
- Output only.
- You must have the Geocode add-on installed to generate this information.

GEOCODE_CENSUS_TRACT

Description

The census tract in which an address is located.

Notes

- A census tract is a geographic area, often coinciding with the limits of cities or towns, defined for census purposes. A census tract can contain several census blocks.
- Output only.
- You must have the Geocode add-on installed to generate this information.

GEOCODE_FOOTNOTE

Description

Indicates the results of Geocode processing.

Notes

- Requires the Geocode add-on.
- Output only.
- Contains one of the following:
  - 00 – Unable to perform a geocode lookup.
  - 03 – Geocode data based on a 3-digit ZIP Code.
  - 05 – Geocode data based on a 5-digit ZIP Code.
  - 07 – Geocode data based on a 7-digit ZIP Code.
  - 09 – Geocode data based on a 9-digit ZIP Code.

GEOCODE_LATITUDE

Description

The latitude of this address.

Notes

- Output only.
- You must have the Geocode add-on installed to generate this information.

GEOCODE_LONGITUDE

Description

The longitude of this address.

Notes

- Output only.
- You must have the Geocode add-on installed to generate this information.

GEOCODE_MSA_CODE

Description

Contains the Metropolitan Statistical Area code for this address.

Notes

- Output only.
- You must have the Geocode add-on installed to generate this information.

IS_RESIDENCE

Description

Indicates whether this address is a residence or not.

Notes

- To use this property, you must have the RDI add-on and subscribe to the RDI service from the USPS.
- Output only.

LACS_CODED

Description

Indicates whether this address was flagged by LACS^{Link} processing.

Notes

Output only.

LACS_FOOTNOTE

Description

Provides additional information about the LACS process.

Notes

- Output only.
- Contains one of the following:
  - blank – Not processed / Seed record.
  - 00 – No match.
  - 09 – Matched to default high-rise address; address not updated.
  - 14 – Match failed to build new address.
  - 92 – Match secondary dropped from input.
  - A – Match success.

LACS_INDICATOR

Description

Indicates the results of a LACS process.

Notes

- Output only.
- Returns one of the following:
  - blank – Not processed.
  - N – Match, but there was a failure to build new address.
  - Y – Match success, new address provided.
  - S – Match with secondary dropped from input.
  - F – Seed record.

LAST_LINE

Description

The last line of the address block.

Notes

You can set or retrieve this field instead of CITY, STATE and ZIP_CODE.

LAST_NAME

Description

The first name of the resident of this address.

Notes

This field will not be processed and will be returned the same as input.

LOT_NUMBER

Description

The line-of-travel number for an address.

Notes

- This field is used in carrier route sorts.
- Output only.

MATCHED_TO_DEFAULT

Description

Indicates whether this address was matched to a default ZIP Code.

Notes

- Default ZIP Codes are only five digits in length.
- Output only.

PMB_NUMBER

Description

The private mailbox number for this address.

Notes

POST_DIRECTIONAL

Description

The post-directional indicator for this address.

Notes

PRE_DIRECTIONAL

Description

The pre-directional indicator for this address.

Notes

RECORD_TYPE

Description

Indicates the type of address that this is.

Notes

Contains one of the following characters:

- S – Street record
- P – Post office box
- R – Rural Route or Highway Contract
- H – High-rise, Building or Apartment
- F – Firm Record
- G – General Delivery
- M – Multi-Carrier Record

SKIPPED_CERTIFY

Description

Indicates whether this address was skipped during CASS processing.

Notes

Output only.

STATE

Description

The state for this address.

Notes

STREET_NAME

Description

The name of the street of this address.

Notes

SUFFIX

Description

The street suffix of this address.

Notes

For example, St., Ave. or Ln.

SUITELINK_FOOTNOTE

Description

Indicates the results of Suite^{Link} processing.

Notes

- Output only.
- Returns one of the following:
  - “” – Was not processed by the SuiteLink engine: the address did not qualify for a lookup within the SuiteLink file. Only default high-rise addresses qualify for a SuiteLink lookup.
  - A – The address was processed and secondary information was added to the resulting address.
  - 00 – The address was processed through the SuiteLink engine, but did not result in a successful match; no secondary information was added.

UNIT_DESIGNATOR

Description

The type of unit for the secondary address data.

Notes

For example, Ste., Apt. or Fl.

UNIT_NUMBER

Description

The unit number for the secondary address data.

Notes

URBANIZATION

Description

The urbanization of this address.

Notes

Puerto Rico only.

ZIPCODE

Description

The ZIP Code of this address.

Notes

MOVEService

MOVEService is a Web service interface to the MOVE Task library. Satori Software provides this interface as an alternative to the COM and .NET interfaces. This service takes a list of addresses, corrects them using a CASS Certified address correction process, matches them against the USPS NCOA$^{\text{Link}}$ database, and returns them to you.

The MOVEService is flexible enough where you can control the amount of information you receive and the format in which it arrives. Afterwards, you can save both the Address Correction Report (PS Form 3553) and the NCOALink Processing Report as a single PDF file.

To create an application that uses the MOVEService to update your records with change-of-address information, you must have a valid MailRoom ToolKit Architect registration key and a registered Move Update account. You will also need Move Update credits for your job. Each address processed uses one credit. Contact your account representative for more information.

MOVEService requires that your implementation be able to store and use cookies. Because the MOVEService processing occurs on a remote server and requires that you make multiple function calls to this server, you need some way of maintaining a connection to the job you are running. Cookies will allow you to maintain this continuity with the server.

When using a .NET client, create an instance of the System.Net.CookieContainer class and attach it to your MoveService object. This creates a place to store the session ID cookie that is returned. The following C# sample code shows this:

```
MoveService mService = new MoveService();
if (mService.CookieContainer == null)
    mService.CookieContainer = new System.Net.CookieContainer();
```

The following sample code shows you how to do this in Java:

```
URL wsdlURL = MoveService.WSDL_LOCATION;
MoveService ms = new MoveService(wsdlURL, MoveService.SERVICE);
MoveServiceSoap move = ms.getMoveUpdateServiceSoap12();

// setup to preserve the session cookie
((BindingProvider)move).getRequestContext().put(BindingProvider._
SESSION_MAINTAIN_PROPERTY, Boolean.TRUE);
```

Overview

The following general procedure should be used to implement the MOVEService:

1. Add a Web reference to the MOVEService. The address of the service is https://ws.satorisoftware.com/Architect/US/Move/MoveService.asmx.
2. Create a MOVEService object.
3. Call PrepareTask() on your registration key. This will return a session ID.
4. Create one or more MoveService.AddressBlock object that contain your address records. An AddressBlock object contains an array of MoveService.Address objects. For each Address object, add at least the fields FIRST_NAME, LAST_NAME, BUSINESS, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, STATE and ZIP_CODE. Add any other fields that you would to pass to the MOVEService.
5. Create arrays of MoveService.Field objects that define the sets of input and output data fields. The input field list should match those you added to the block of addresses in the previous step. The output field list can be any or all of the input fields, plus any of the output data fields that the MOVEService produces. See the Fields section of this document for the full list.
6. Call SetProperties to set the desired properties. Add the individual properties to set to a MoveService.MoveServiceProperties enumeration. You must set SETTINGS_FIELD_LIST_IN, SETTINGS_FIELD_LIST_OUT, LOGIN_CUSTOMER_ID, LOGIN_CUSTOMER_PASSWORD and SETTINGS_INPUT_BLOCK_RECORD_COUNT.
7. Call ValidateProperties.
8. Call Send on the AddressBlock object or objects you created in step 4.
9. Call DoProcess().
10. Call GetProcessStatus() to check the state of the process. While the state.phase is Processing, wait.
11. When the state.phase is Complete, call Retrieve. This will return an updated AddressBlock object.
12. Iterate through the AddressBlock object to get each updated Address object. In turn, iterate through each Address object to get the updated fields for each record. The fields for each output record will match those specified by the SETTINGS_FIELD_LIST_OUT property set earlier.
13. Repeat steps 11 and 12 until you have retrieved all records.
14. Call GetReportsAsPDF() to return the CASS Summary Report (PS Form 3553) and NCOALink Processing Summary Report as a byte array, which you can save on your local system.
15. Call EndTask and clean up any other resources used.

MOVEService Functions

Below are the methods, properties and fields available in MOVEService. If you have added a Web reference to the MOVEService in Visual Studio, you can view all of these functions in the Object Browser.

Note that in all the functions below, we use the namespace MoveServiceReference. This may be different on your system, depending on the development environment.

DoProcess

Syntax

```
void DoProcess();
```

Description

Starts processing all the addresses that you have sent to the MOVEService.

Parameters

None.

Return values

None.

Notes

- This function does not return anything, as processing may take a significant amount of time.
- Call GetProcessStatus() to return the current state of the process.

See also

GetProcessStatus()

EndTask

Syntax

```
void EndTask();
```

Description

Formally ends the MOVEService process and cleans up any resources that it used.

Parameters

None.

Return values

None.

Notes

Call this when you are finished processing addresses and you have called Retrieve() for all of your blocks of addresses.

See also

Retrieve()

GetProcessStatus

Syntax

```
ProcessingStatus GetProcessStatus();
```

Description

Returns the current state of the MOVEService processing.

Parameters

None.

Return values

A ProcessingStatus object. This object contains two values, phase and recordsProcessed. The recordsProcessed variable is an integer that indicates how many records have been processed. The phase object contains a MoveServiceReference.ProcessingState object, which will be one of the following:

- Processing – The process is still in progress, so please wait and get the status again.
- Complete – The process is now finished and you can call Retrieve() to get the updated addresses.
- Error – The MOVEService was unable to complete the process.

Notes

- While this function returns Processing as the phase value, do not call Retrieve(). Doing so will return an error.

See also

- Retrieve()

GetProperties

Syntax

```
MoveServiceReference.MoveServiceProperties
GetProperties(MoveServiceReference.ArrayofProperty
PropertyIDs);
```

Description

Returns the values of the Property items passed to it.

Parameters

A Property array that contains the properties for which you want to get the values.

Return values

A MoveServiceProperties object that contains the properties specified by the Property array passed to it.

Notes

- The returned object will only contain those properties specified in the parameter array.
- Any properties that you have not already set will return the default values.

See also

- ValidateProperties()
- SetProperties()
- The MoveServiceProperties list.

GetReportPDF

Syntax

```
byte[] GetReportPDF();
```

Description

Returns the Address Correction Report and NCOALink Processing Report as a byte array that can be saved on the local machine as a PDF file.

Parameters

None.

Return values

A byte array that can saved on the local machine as a PDF file.

Notes

To open a PDF file, use Adobe Reader, freely available from the Web site of Adobe Software.

See also

PrepareTask

Syntax

```
string PrepareTask(string RegistrationKey);
```

Description

Establishes a session with the MOVEService using your registration key.

Parameters

Your MailRoom ToolKit Architect registration key as a string.

Return values

A session ID as a string.

Notes

- This method must be called before you can set any properties or process any addresses.
- Call this function only once, after you create the MOVEService object.
- You may need your session ID if you are disconnected from the service. Call ReconnectTask() to reestablish a connection with the MOVEService.

See also

ReconnectTask()

Retrieve

Syntax

```
MoveServiceReference.AddressBlock Retrieve();
```

Description

Returns the updated addresses once processing is complete.

Parameters

None.

Return values

An AddressBlock object that contains processed addresses. The AddressBlock object contains an array of Address objects.

Notes

- The addresses in the AddressBlock object will have all the fields specified in the SETTINGS_FIELD_LIST_OUT property.
- Each address block will contain a number of addresses specified by the property, SETTINGS_INPUT_BLOCK_RECORD_COUNT.
- You may need to call this function several times to return all of your processed addresses.

See also

- SETTINGS_FIELD_LIST_OUT
- SETTINGS_INPUT_BLOCK_RECORD_COUNT
- Send()

Send

Syntax

```
void Send(AddressBlock AddressBlock);
```

Description

Sends the specified block of addresses to the MOVEService to be processed.

Parameters

An AddressBlock object that contains the addresses you wish to process.

Return values

None.

Notes

- You can send up to 500 addresses in each address block. You may want to send fewer to improve processing speed.
- Call this function for every block of addresses you wish to process.
- Call DoProcess() to start processing these addresses.

See also

- DoProcess()
- Retrieve()

SetProperties

Syntax

```
void
SetProperties(MoveServiceReference.MoveServiceProperties
MoveProps);
```

Description

Sets the properties for the MOVEService process using a MoveServiceProperties enumeration.

Parameters

A MoveServiceProperties object that contains the settings you would like to apply to this process.

Return values

None.

Notes

- You can set as many properties at once as you wish.
- You must set SETTINGS_FIELD_LIST_IN, SETTINGS_FIELD_LIST_OUT, LOGIN_CUSTOMER_ID, LOGIN_CUSTOMER_PASSWORD and SETTINGS_INPUT_BLOCK_RECORD_COUNT.
- Once you have set the properties, you must then call ValidateProperties().

See also

- ValidateProperties()
- GetProperties()
- The MoveServiceProperties list.

ValidateProperties

Syntax

```
Bool ValidateProperties();
```

Description

Validates the properties that you have passed to the service.

Parameters

None.

Return values

Returns true if the properties sent are valid.

Notes

- You must call this function before DoProcess().
- This function ensures that you have set all of the necessary properties.

See also

- SetProperties()
- GetProperties()
- The MoveServiceProperties list.

MoveServiceProperties

Use the following properties to either configure the MOVEService using SetProperties(MoveServiceProperties) or to retrieve the existing settings using GetProperties(MoveServiceProperties). The MoveServiceProperties is an enumeration that can contain any or all or these properties.

Below the name of the property is the data type it takes or returns. All properties can be set or retrieved. Any properties that you attempt to retrieve before you assign a value to them will return the default value. When processing addresses with the MOVEService, you must set SETTINGS_FIELD_LIST_IN, SETTINGS_FIELD_LIST_OUT, LOGIN_CUSTOMER_ID, LOGIN_CUSTOMER_PASSWORD and SETTINGS_INPUT_BLOCK_RECORD_COUNT.

Note that in all the properties below, we use the namespace MoveServiceReference. This may be different on your system, depending on the development environment.

CASS_DPV_FAILURE_AS_ERROR

Data Type

string

Description

Determines whether the service treats DPV failures that would otherwise allow an address to receive a ZIP + 4 Code as errors.

Notes

- While DPV is required for CASS processing, addresses that fail DPV because of missing or invalid secondary information will still pass CASS processing.
- Set this property to 1 to enable it.

CASS_DUAL_ADDRESS_INPUT_PREFERENCE

Data Type

string

Description

Determines whether to give the street address or PO Box preference for addresses that have both.

Notes

Use the following to set this property:

- 0 – The bottom address will be used, regardless of which type it is.
- 1 – Uses PO Box addresses.
- 2 – Uses street addresses.

CASS_KEEP_ALIAS_ADDRESS

Data Type

string

Description

Determines whether an input address with a valid street name alias will be allowed or whether they will be replaced with the official USPS street name.

Notes

Set this property to 1 to enable it.

CASS_KEEP_EXTRA_PRIMARY_DATA

Data Type

string

Description

Determines whether to keep any additional and possibly extraneous information in the primary address line.

Notes

Set this property to 1 to enable it.

CASS_LIST_NAME

Data Type

string

Description

The name of the list being processed.

Notes

This name will print on the Address Correction report.

CASS_LIST_PROCESSOR

Data Type

string

Description

The name of the person processing the list.

Notes

This will be printed on the Address Correction Report.

CASS_MAILERS_ADDRESS

Data Type

string

Description

The address of the company or individual doing the mailing.

Notes

This will be printed on the Address Correction Report.

CASS_MAILERS_CITY

Data Type

string

Description

The city of the company or individual doing the mailing.

Notes

This will be printed on the Address Correction Report.

CASS_MAILERS_NAME

Data Type

string

Description

The name of the company or individual doing the mailing.

Notes

This will be printed on the Address Correction Report.

CASS_MAILERS_STATE

Data Type

string

Description

The state in which the company or individual doing the mailing resides.

Notes

This will be printed on the Address Correction Report.

CASS_MAILERS_ZIP

Data Type

string

Description

The ZIP Code of the company or individual doing the mailing.

Notes

This will be printed on the Address Correction Report.

CASS_UPDATE_UNCORRECTED_CITY_ST_ZIP

Data Type

boolean

Description

Determines whether the corrected city, state and/or ZIP Code should be returned for addresses that otherwise could not be corrected.

Notes

FORMAT_CASING

Data Type

Capitalization

Description

Sets the capitalization format for processed addresses.

Notes

Use one of the following:

- MoveServiceReference.Capitalization.eCapUpper – Sets to all UPPER CASE.
- MoveServiceReference.Capitalization.eCapLower – Sets to all lower case.
- MoveServiceReference.Capitalization.eCapMixed – Sets to Mixed Case.

FORMAT_CITY

Data Type

AbbreviateCity

Description

Determines whether to abbreviate the city returned from the MOVEService process.

Notes

Use one of the following:

- MoveServiceReference.AbbreviateCity.eCityAbbrNever – The city will not be abbreviated.
- MoveServiceReference.AbbreviateCity.eCityAbbrForce – The city will always be abbreviated, if possible.
- MoveServiceReference.AbbreviateCity.eCityAbbrInput – The output city will be abbreviated only if the input city was.

FORMAT_FIRM_OUTPUT

Data Type

FirmOutput

Description

Determines how to return a firm name found with an address record.

Notes

Use one of the following:

- MoveServiceReference.FirmOutput.eFirmMoveToBusiness – If the input BUSINESS field was blank, return field information in the BUSINESS field.
- MoveServiceReference.FirmOutput.eFirmMoveToBusinessOverwrite – Always output the firm information as the BUSINESS field.

FORMAT_HIGHWAY_CONTRACT

Data Type

AddressElementFormat

Description

Determines how the service formats Highway Contract address information.

Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

FORMAT_PMB_OUTPUT

Data Type

PMBOutput

Description

Determines the output location of personal mailbox information, if any.

Notes

Use one of the following:

- MoveServiceReference.PMBOutput.ePMBWithUnits – PMB information outputs on the same line as the unit information.

FORMAT_PO_BOX

Data Type

AddressElementFormat

Description

Determines how the service will output PO Box information.

Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

FORMAT_POST_DIRECTIONAL

Data Type

AddressElementFormat

Description

Determines how the service will format post-directional information.

Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

FORMAT_PRE_DIRECTIONAL

Data Type

AddressElementFormat

Description

Determines how the service will format pre-directional information.

Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

FORMAT_PRIMARY_ADDRESS_OUTPUT

Data Type

PrimaryAddOutput

Description

Determines how the service will output the primary address information.

Notes

Set this to one of the following:

- MoveServiceReference.PrimaryAddOutput.eTopJustified – Returns primary address data as ADDRESS_LINE_1 unless the input address contains secondary information in ADDRESS_LINE_2. In this case, the secondary information will be returned as ADDRESS_LINE_1 and the primary address data will be returned in ADDRESS_LINE_2.
- MoveServiceReference.PrimaryAddOutput.eAddressLine2 – Always returns the primary address data as ADDRESS_LINE_2.

FORMAT_RURAL_ROUTE

Data Type

AddressElementFormat

Description

Determines how the service will format rural route address information.

Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

FORMAT_SUFFIX

Data Type

AddressElementFormat

Description

Determines how the service will format street suffix information.

Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

FORMAT_UNIT_DESIGNATOR

Data Type

AddressElementFormat

Description

Determines how the service will format unit type information, such as Suite or Apartment.

Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

FORMAT_UNIT_OUTPUT

Data Type

UnitOutput

Description

Determines where the service outputs the secondary address information, such as the unit or apartment number.

Notes

- In the event this conflicts with FORMAT_PRIMARY_ADDRESS_OUTPUT, the primary format will take precedence.
- Set this to one of the following:
  - MoveServiceReference.UnitOutput.ePrimaryAddressLine – Appends this data to the primary address information. This is the default for this property.
  - MoveServiceReference.UnitOutput.eSecondaryAddressLine – Returns this data as ADDRESS_LINE_2 only if that field was blank.
  - MoveServiceReference.UnitOutput.eSecondaryAddressLineOverwrite – Always returns this data as ADDRESS_LINE_2 and overwrites any existing data there.

FORMAT_UPDATE_CASE_BUSINESS

Data Type

boolean

Description

Determines whether to apply the casing specified by FORMAT_CASING to the BUSINESS field.

Notes

FORMAT_UPDATE_CASE_NAMES

Data Type

boolean

Description

Determines whether to apply the casing specified by FORMAT_CASING to the FIRST_NAME and LAST_NAME fields.

Notes

LOGIN_ADMIN_ID

Data Type

string

Description

The login ID if you have an administrator account.

Notes

In order to run the MOVEService if you have a list administrator account, you must set this property to the login ID that Satori Software has supplied to you.

LOGIN_ADMIN_PASSWORD

Data Type

string

Description

The password if you have an administrator account.

Notes

In order to run the MOVEService if you have a list administrator account, you must set this property to the password that Satori Software has supplied to you.

LOGIN_BROKER_ID

Data Type

string

Description

The login ID if you have a list broker account.

Notes

In order to run the MOVEService if you have a list broker account, you must set this property to the login ID that Satori Software has supplied to you.

LOGIN_BROKER_PASSWORD

Data Type

string

Description

The password if you have a list broker account.

Notes

In order to run the MOVEService if you have a list broker account, you must set this property to the password that Satori Software has supplied to you.

LOGIN_CUSTOMER_ID

Data Type

string

Description

The login ID if you have a customer account.

Notes

In order to run the MOVEService if you have a client account, you must set this property to the login ID that Satori Software has supplied to you.

LOGIN_CUSTOMER_PASSWORD

Data Type

string

Description

The password if you have a customer account.

Notes

In order to run the MOVEService if you have a client account, you must set this property to the password that Satori Software has supplied to you.

MOVE_BUYER_NAME

Data Type

string

Description

The name of the client who owns the list being processed.

Notes

This property only applies to administrator or list broker accounts.

MOVE_CLIENT_ID_LIST

Data Type

String

Description

Returns a list of client IDs and their PAF expiration dates as a single string.

Notes

- The format is [CLIENT_ID],YYYY-MM-DD. Each ID-date pair is separated from others by a carriage return.
- Before you can retrieve this data, you must set LOGIN_BROKER_ID or LOGIN_CUSTOMER_ID first, then call ValidateProperties.
- The default value is an empty string.

MOVE_CUSTOMER_MAILERID

Data Type

string

Description

Your mailer ID that the USPS assigned to you.

Notes

This will be printed on the NCOALink Processing Form.

MOVE_HIGH_MATCH_RATE_REASON

Data Type

string

Description

If you have a mailing list that produces an unusually high rate of matches to changed address entries, then you should provide a reason why this occurs.

Notes

- You will only need to set this property if your list was previously processed by ANK or another similar tool.
- This text will be printed on the NCOALink Processing Report if necessary.

MOVE_MAIL_CLASS

Data Type

string

Description

The mail class by which you intend to send the processed mailing.

Notes

Use the following:

- A – First Class Only
- B – Periodicals Only
- C – Standard Mail Only
- D – Package Services Only
- E – First Class And Periodicals
- F – First Class And Standard
- G – First Class And Package
- H – Periodicals And Standard
- I – Periodicals And Package
- J – Standard And Package
- K – First Class, Periodicals, Standard
- L – First Class, Periodicals, Package
- M – First Class, Standard, Package
- N – Periodicals, Standard, Package

- O – All (default)

MOVE_MATCH_FLAG

Data Type

string

Description

Determines what type of moves the service will search for.

Notes

Use one of the following:

- S – All move types: Business, Individual and Family
- C – Business and Individual
- B – Business only
- I – Individual only

MOVE_MOVE_MONTH_RANGE

Data Type

string

Description

Determines how the maximum age of the move for which the service will search.

Notes

- You may search for moves that are up to 48 months old.
- Set this property to the maximum number of months in which to search.

MOVE_MULTI_NAME_HANDLE

Data Type

string

Description

Determines how the service treats records with multiple names.

Notes

Use one of the following:

- 0 – Search only if a common last name is found.
- 1 – Search using all names found.
- 2 – Skip the address.

SETTINGS_FIELD_LIST_IN

Data Type

ArrayOfField

Description

An array of MoveServiceReference.Field items that identify which fields the input addresses will contain.

Notes

- This is required to run the MOVEService.
- You must add at least ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, STATE and ZIP_CODE.
- Depending on the type of moves you are searching for, you will also need one or more of the following: FIRST_NAME, LAST_NAME and BUSINESS.
- See the fields below for the full list of available Fields.

SETTINGS_FIELD_LIST_OUT

Data Type

ArrayOfField

Description

An array of MoveServiceReference.Field items that determine which fields the processed output addresses will contain.

Notes

- This is required to run the MOVEService.
- We recommend that you add at least ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, STATE and ZIP_CODE.
- See the fields below for the full list of available Fields.

SETTINGS_INPUT_BLOCK_RECORD_COUNT

Data Type

int

Description

Set this equal to the number of addresses you send per block of addresses.

Notes

- Each address block should contain this number of records, except for the last block, which can contain less.
- You must set this property in order to run a MOVEService job.

SETTINGS_RECORD_COUNT

Data Type

int

Description

Set this equal to the total number of records in your mailing list.

Notes

This property is required if you send multiple blocks of addresses.

Fields

The items below are part of the Field enumeration. Include these names in the ArrayOfFields passed to either SETTINGS_FIELD_LIST_IN or SETTINGS_FIELD_LIST_OUT to determine which fields MOVEService looks to process from incoming addresses or returns in processed records. All field values are strings.

Many of these fields are output only. The descriptions below will indicate which fields do not accept input. All fields with the prefixes BEFORE_CASS, AFTER_CASS or AFTER_NCOA do not accept input.

For an effective MOVEService process, we recommend that you use at least the following fields as input: ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, STATE and ZIP_CODE. Depending on the type of moves you are searching for, you will also need one of more of the following: FIRST_NAME, LAST_NAME and BUSINESS.

Note that in all the fields below, we use the namespace MoveServiceReference. This may be different on your system, depending on the development environment.

The following fields must be in the list of input fields if you wish to include them in the list of output fields:

- RECORD_ID
- All USER_DEFINED fields
- COUNTRY
- NAME_SALUTATION
- MIDDLE_NAME
- NAME_SUFFIX

RECORD_ID

Description

A number that identifies a record.

Notes

- This will not be generated during a MOVEService process.
- You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.

FIRST_NAME

Description

The first name of the individual associated with this record.

Notes

- This will be used for Individual matches.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_FIRST_NAME
  - AFTER_NCOA_MATCH_FIRST_NAME – This is the first name of the matched record in the NCOALink database.

LAST_NAME

Description

The last name of the individual associated with this record.

Notes

- This will be used for Individual and Family matches.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_LAST_NAME
  - AFTER_NCOA_MATCH_LAST_NAME – This is the last name of the matched record in the NCOALink database.

NAME_SALUTATION

Description

A greeting (Mr., Mrs, Ms.) for the name on this record.

Notes

- You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.
- Has the following related field that shows this data before CASS processing:
  - BEFORE_CASS_PREFIX_TITLE

MIDDLE_NAME

Description

The middle name of the person on this record.

Notes

- You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_MIDDLE_NAME
  - AFTER_NCOA_MIDDLE_INITIAL

NAME_SUFFIX

Description

The suffix (Jr., Sr., etc.), if any, for the person on this record.

Notes

- You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_SUFFIX_TITLE
  - AFTER_NCOA_MATCH_SUFFIX_NAME

BEFORE_CASS_FULL_NAME

Description

The full name of this record before CASS processing.

Notes

This is assembled from the FIRST_NAME and LAST_NAME fields.

BUSINESS

Description

The name of the business associated with this address.

Notes

- This will be used for Business matches.
- Has the following related field that shows this data before CASS processing:
  - BEFORE_CASS_BUSINESS

ADDRESS_LINE_1

Description

The first line of this address.

Notes

- This will be the primary address information unless you have unit information on a separate line.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_ADDRESS_LINE_1
  - AFTER_CASS_ADDRESS_LINE_1
  - AFTER_NCOA_ADDRESS_LINE_1

ADDRESS_LINE_2

Description

The second line of this address.

Notes

- If you have unit information on a separate line, this may contain the street address.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_ADDRESS_LINE_2
  - AFTER_CASS_ADDRESS_LINE_2
  - AFTER_NCOA_ADDRESS_LINE_2

CITY

Description

The city for this address.

Notes

Has the following related fields that show this data before and after various processes:

- BEFORE_CASS_CITY
- AFTER_CASS_CITY
- AFTER_NCOA_CITY

STATE

Description

The state for this address.

Notes

Has the following related fields that show this data before and after various processes:

- BEFORE_CASS_STATE
- AFTER_CASS_STATE
- AFTER_NCOA_STATE

ZIP_CODE

Description

The ZIP Code for this address.

Notes

- During CASS processing, successfully corrected addresses will receive a valid ZIP + 4 Code in this field.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_ZIP_CODE
  - AFTER_CASS_ZIP_CODE
  - AFTER_NCOA_ZIP_CODE
  - AFTER_NCOA_QUERY_ZIP_CODE

LAST_LINE

Description

Contains the city, state and ZIP Code for this address.

Notes

Has the following related fields that show this data before and after various processes:

- BEFORE_CASS_LAST_LINE
- AFTER_CASS_LAST_LINE
- AFTER_NCOA_LAST_LINE

COUNTY_NAME

Description

The name of the county in which this address is located.

Notes

Has the following related fields that show this data before and after various processes:

- AFTER_CASS_COUNTY_NAME
- AFTER_NCOA_COUNTY_NAME

COUNTY_CODE

Description

The code for the county in which this address is located.

Notes

Has the following related fields that show this data before and after various processes:

- AFTER_CASS_COUNTY_CODE
- AFTER_NCOA_COUNTY_CODE

URBANIZATION

Description

The urbanization for this address.

Notes

- This field applies to Puerto Rico addresses only.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_URBANIZATION
  - AFTER_CASS_URBANIZATION
  - AFTER_NCOA_URBANIZATION

COUNTRY

Description

The country of this address.

Notes

CONGRESSIONAL_DISTRICT

Description

Returns the congressional district of this address.

Notes

- Output only.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_CONGRESSIONAL_DISTRICT
  - AFTER_NCOA_CONGRESSIONAL_DISTRICT

ADDRESS_BLOCK

Description

The full address as would be printed on the front of a mail piece.

Notes

Has the following related fields that show this data before and after various processes:

- AFTER_CASS_ADDRESS_BLOCK
- AFTER_NCOA_ADDRESS_BLOCK

EXTRA_INFO

Description

Contains extra address information that the service was unable to parse into any other fields.

Notes

Output only.

CASSDATE

Description

An encoded string that contains the date that this record was last processed.

Notes

- Output only.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_CASSDATE
  - AFTER_NCOA_CASSDATE

ERROR_CODE

Description

A numerical code that indicates the results of CASS processing.

Notes

- Output only.
- See the appendix for the codes.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_ERROR_CODE
  - AFTER_NCOA_ERROR_CODE

ERROR_STRING

Description

Text that explains the results from CASS processing.

Notes

- Output only.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_ERROR_STRING
  - AFTER_NCOA_ERROR_STRING

LOT_NUMBER

Description

The line-of-travel number for this address.

Notes

- Output only.
- Returned from CASS processing.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_LOT_NUMBER
  - AFTER_NCOA_LOT_NUMBER

CARRIER_ROUTE

Description

The carrier route for this address.

Notes

- Output only.
- This information is returned during CASS processing, which is performed during the MOVEService process.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_CARRIER_ROUTE
  - AFTER_CASS_CARRIER_ROUTE
  - AFTER_NCOA_CARRIER_ROUTE

DPC

Description

Contains the delivery point code for this address.

Notes

- Output only.
- This data is generated during CASS processing.
- Has the following related fields that show this data before and after various processes:
  - BEFORE_CASS_DPC
  - AFTER_CASS_DPC
  - AFTER_NCOA_DPC

DP_BARCODE

Description

The delivery point barcode for this address record.

Notes

- Output only.
- This data will be generated during CASS processing.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_DP_BARCODE
  - AFTER_NCOA_DP_BARCODE

USER_DEFINED fields

Description

Custom data to attach to a record.

Notes

There are 15 user defined fields for you to use, named USER_DEFINED_1 to USER_DEFINED_15

BEFORE_CASS_PRIMARY_NUMBER

Description

The street number for the address as it was before CASS processing.

Notes

- This information is parsed from ADDRESS_LINE_1 and ADDRESS_LINE_2.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_PRIMARY_NUMBER
  - AFTER_NCOA_PRIMARY_NUMBER

BEFORE_CASS_PRE_DIRECTIONAL

Description

The pre-directional, if any, as it was before CASS processing.

Notes

- This information is parsed from ADDRESS_LINE_1 and ADDRESS_LINE_2.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_PRE_DIRECTIONAL
  - AFTER_NCOA_PRE_DIRECTIONAL

BEFORE_CASS_POST_DIRECTIONAL

Description

The post-directional information, if any, as it was before CASS processing.

Notes

- This information is parsed from ADDRESS_LINE_1 and ADDRESS_LINE_2.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_POST_DIRECTIONAL
  - AFTER_NCOA_POST_DIRECTIONAL

BEFORE_CASS_STREET_NAME

Description

The name of the street as it was before CASS processing.

Notes

- This information is parsed from ADDRESS_LINE_1 and ADDRESS_LINE_2.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_STREET_NAME
  - AFTER_NCOA_STREET_NAME

BEFORE_CASS_SUFFIX

Description

The street suffix as it was before CASS processing.

Notes

- This information is parsed from ADDRESS_LINE_1 and ADDRESS_LINE_2.

- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_SUFFIX
  - AFTER_NCOA_SUFFIX

BEFORE_CASS_UNIT_NUMBER

Description

The unit number as it was before CASS processing.

Notes

- This information is parsed from ADDRESS_LINE_1 and ADDRESS_LINE_2.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_UNIT_NUMBER
  - AFTER_NCOA_UNIT_NUMBER

BEFORE_CASS_UNIT_DESIGNATOR

Description

The unit type designator as it was before CASS processing.

Notes

- This information is parsed from ADDRESS_LINE_1 and ADDRESS_LINE_2.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_UNIT_DESIGNATOR
  - AFTER_NCOA_UNIT_DESIGNATOR

AFTER_CASS_PMB_NUMBER

Description

The postal mailbox number, if any, of the address after being CASS processed.

Notes

Has a related field for this data after NCOA processing.

- AFTER_NCOA_PMB_NUMBER

ZIP4_FOOTNOTE

Description

Returns information about the CASS process.

Notes

- Output only.
- This field can be one of the following:
  - AA – Input Address ZIP + 4 match
  - A1 – Input Address ZIP + 4 not matched
  - M1 – Input Address Primary Number Missing
  - M3 – Input Address Primary Number Invalid
  - P1 – Input Address Missing PO, RR or HC Box number
  - P3 – Input Address Invalid PO, RR or HC Box number
  - BB – Input Address DPV matched (all components)
  - RR – Input Address DPV matched to CMRA
  - CC – Input Address DPV Primary Number match, Secondary Number not Matched (secondary number present but is not DPV confirmed)
  - N1 – Input Address DPV Primary Number match, High-rise Address Missing Secondary Number
  - R1 – Input Address DPV matched to CMRA but PMB Number not Present

DPV_CODED

Description

Indicates whether an address was verified using DPV.

Notes

- Output only.
- For an address to receive a valid ZIP + 4 Code during CASS processing, it must be verified by DPV.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_DPV_CODED
  - AFTER_NCOA_DPV_CODED

DPV_IS_CMRA

Description

Indicates whether this address is a commercial mail-receiving agent.

Notes

- Output only.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_DPV_IS_CMRA
  - AFTER_NCOA_DPV_IS_CMRA

DPV_IS_VACANT

Description

Indicates whether this address has been vacant for at least 90 days.

Notes

- Output only.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_DPV_IS_VACANT
  - AFTER_NCOA_DPV_IS_VACANT

DPV_INDICATOR

Description

Returns a single character that describes the result of DPV processing.

Notes

- Output only.
- Returns one of the following:
  - Y – Both the primary and secondary address data validated against the DPV database.
  - S – The primary address is valid, but the secondary is invalid.
  - D – The primary address is valid, but the address is missing secondary information.
  - N – The primary address is not valid. This address was not given a ZIP + 4 Code.
  - “” – The address was not presented to the DPV table because it was missing components needed for the lookup.
  - X – The DPV database has been locked-out because of a protocol violation. You must unlock DPV before any more addresses can be processed with DPV.
  - E – The DPV data file is more than 105 days old. By USPS restrictions, no more addresses can be presented to the DPV table.

- Has the following related fields that show this data before and after various processes:
    - AFTER_CASS_DPV_INDICATOR
    - AFTER_NCOA_DPV_INDICATOR

DPV_FOOTNOTE

Description

Indicates the results of the DPV verification process.

Notes

- Output only.
- Returns one of the following:
    - LK – Processing locked out due to a seed record being processed
    - AA – Matched to the ZIP+4 file
    - A1 – No match against the ZIP+4 file
    - BB – Matched to DPV file (all components confirmed)
    - CC – Matched only after removing secondary Information; they were presented but invalid
    - N1 – Input Primary matched, but high-rise missing secondary number
    - M1 – Primary number missing
    - M3 – Primary number invalid
    - P1 – Input missing PO, RR, HC box number
    - P3 – Failed DPV because of invalid PO, RR, or HC box number
    - RR – Matched CMRA (found in CMRA file)
    - R1 – Matched CMRA, but secondary number (i.e., PMB) missing
    - U1 – Matched unique zip code
    - G1 – Matched General delivery
    - F1 – Matched military address

LACS_CODED

Description

Indicates if this address was changed due to a match in the LACS database.

Notes

- Output only.
- LACS processing occurs as part of CASS processing.
- Has the following related fields that show this data before and after various processes:
    - AFTER_CASS_LACS_CODED

- AFTER_NCOA_LACS_CODED

EWS_CODED

Description

Indicates that this record was flagged by the early warning system as a new address.

Notes

- Output only.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_EWS_CODED
  - AFTER_NCOA_EWS_CODED

RECORD_TYPE

Description

The type of address record this is.

Notes

- Output only.
- Returns one of the following characters:
  - S – Street record
  - P – Post office box
  - R – Rural Route or Highway Contract
  - H – High-rise, Building or Apartment
  - F – Firm Record
  - G – General Delivery
  - M – Multi-Carrier Record
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_RECORD_TYPE
  - AFTER_NCOA_RECORD_TYPE

MATCHED_TO_DEFAULT

Description

This field will return 1 if CASS processing has matched it to a default ZIP Code.

Notes

- Output only.
- Default ZIP Codes are for large, single addresses, such as universities or corporations, that do not have +4 codes.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_MATCHED_TO_DEFAULT
  - AFTER_NCOA_MATCHED_TO_DEFAULT

MOVE_EFFECTIVE

Description

The date that the matched move became or becomes the record’s active address.

Notes

Output only.

MOVE_TYPE

Description

The type of move found.

Notes

- Output only.
- This returns one of the following:
  - I – Individual
  - F – Family
  - B – Business

MATCH_FLAG

Description

Data about the match that the MOVEService made with this address.

Notes

- Output only.
- Returns one of the following:

- M – Matched; updated address
- F – Foreign Move; new address unavailable
- K – No forwarding address; new address unavailable
- G – PO box closed; new address unavailable
- N – No match
- X – Other

MOVE_FOOTNOTE

Description

Describes the results of the MOVEService processing.

Notes

- Output only.
- This field will contain one of the following:

Match Found – new address returned

- A – Input record matched
- 91 – Secondary number dropped from change of address
- 92 – Secondary number dropped from input address

Match Found – new address unavailable

- 1 – Foreign move
- 2 – Move left no address
- 3 – PO box closed; no forwarding
- 5 – New 11-digit DPBC is ambiguous
- 14 – New address would not convert to deliverable
- 19 – New address not ZIP + 4 coded

No Match Found

- 00 – No move found
- 4 – Street address missing secondary
- 6 – Conflicting directions, middle name related
- 7 – Conflicting directions, gender related
- 8 – Other conflicting instructions
- 9 – High-rise default
- 10 – Rural route default
- 11 – Individual, insufficient name for match
- 12 – Middle name test failed
- 13 – Gender test failed
- 15 – Individual name insufficient
- 16 – Secondary number discrepancy
- 17 – Other insufficient name
- 18 – General delivery
- 20 – Conflicting directions after rechaining
- 66 – Address deleted, no forwarding allowed

MOVE_FOOTNOTE_SHORT_DESCRIPTION

Description

Returns a short description of the results of the MOVEService processing.

Notes

Output only.

MOVE_FOOTNOTE_LONG_DESCRIPTION

Description

Returns a longer, more detailed description of the results of the MOVEService processing.

Notes

Output only.

LACS_FOOTNOTE

Description

Provides additional information about the LACS process.

Notes

- Output only.
- Returns one of the following:
  - blank – Not processed / Seed record.
  - 00 – No match.
  - 09 – Matched to default high-rise address; address not updated.
  - 14 – Match failed to build new address.
  - 92 – Match secondary dropped from input.
  - A – Match success.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_LACS_FOOTNOTE
  - AFTER_NCOA_LACS_FOOTNOTE

LACS_INDICATOR

Description

Indicates the results of a LACS process.

Notes

- Output only.
- Returns one of the following:
  - blank – Not processed.
  - N – Match, but there was a failure to build new address.
  - Y – Match success, new address provided.
  - S – Match with secondary dropped from input.
  - F – Seed record.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_LACS_INDICATOR
  - AFTER_NCOA_LACS_INDICATOR

SUITELINK_FOOTNOTE

Description

Indicates the results of SuiteLink matching.

Notes

- Output only.
- Returns one of the following:
  - blank – Was not processed by the SuiteLink engine: the address did not qualify for a lookup. Only default high rise addresses qualify for a SuiteLink lookup.
  - A – The address was processed and secondary information was added to the resulting address.
  - 00 – The address was processed through the SuiteLink engine, but did not result in a successful match; no secondary information was added.
- Has the following related fields that show this data before and after various processes:
  - AFTER_CASS_SUITELINK_FOOTNOTE
  - AFTER_NCOA_SUITELINK_FOOTNOTE

Appendix

CASS Error Codes

The error codes below indicate either the results of an attempt to correct an address as stored in the ERROR_CODE field. Codes 0 – 99 are considered successful.

VALUE	DESCRIPTION
Corrected	
0, 7, 8	No Changes
0	No significant changes
7	Matched to unique ZIP code; delivery address line not verified
8	Matched to general delivery ZIP code; delivery address line not verified
9, 10	*LACS/SuiteLink Changes*
9	Address revised by LACS
10	Secondary information added by SuiteLink
11-20	Special Situations
11	Alias match
12	ZIP move match
13	Alternate match
21-30	Last Line Changes
21	City changed
22	City and state changed
23	City and ZIP changed
24	City, state, and ZIP changed
25	ZIP changed
26	ZIP and State changed
27	State changed
31-99	Delivery Address Line Changes
31	Street name changed
32	Street name and suffix changed
33	Street name and Directional changed
34	Street name, suffix, and directional changed
35	Street name and city changed
36	Street name, city, and state changed
37	Street name, city, and ZIP changed
38	Street name, city, state, and ZIP changed
39	Street name and ZIP changed
40	Street name, ZIP, and state changed
41	Street name and state changed

42	Street name, suffix, and city changed
43	Street name, suffix, city, and state changed
44	Street name, suffix, city, and ZIP changed
45	Street name, suffix, city, state, and ZIP changed
46	Street name, suffix, and ZIP changed
47	Street name, suffix, ZIP, and state changed
48	Street name, suffix, and state changed
49	Street name, directional, and city changed
50	Street name, directional, city, and state changed
51	Street name, directional, city, and ZIP changed
52	Street name, directional, city, state, and ZIP changed
53	Street name, directional, and ZIP changed
54	Street name, directional, ZIP, and state changed
55	Street name, directional, and state changed
56	Street name, suffix, directional, and city changed
57	Street name, suffix, directional, city, and state changed
58	Street name, suffix, directional, city, and ZIP changed
59	Street name, suffix, directional, city, state, and ZIP changed
60	Street name, suffix, directional, and ZIP changed
61	Street name, suffix, directional, ZIP, and state changed
62	Street name, suffix, directional, and state changed
63	Suffix changed
64	Suffix and directional changed
65	Suffix and city changed
66	Suffix, city, and state changed
67	Suffix, city, and ZIP changed
68	Suffix, city, state, and ZIP changed
69	Suffix and ZIP changed
70	Suffix, ZIP, and state changed
71	Suffix and state changed
72	Suffix, directional, and city changed
73	Suffix, directional, city, and state changed
74	Suffix, directional, city, and ZIP changed
75	Suffix, directional, city, state, and ZIP changed
76	Suffix, directional, and ZIP changed
77	Suffix, directional, ZIP, and state changed
78	Suffix, directional, and state changed
79	Directional changed
80	Directional and city changed
81	Directional, city, and state changed
82	Directional, city, and ZIP changed
83	Directional, city, state, and ZIP changed
84	Directional and ZIP changed
85	Directional, ZIP, and state changed
86	Directional and state changed
91-99	DPV Warnings
92	Failed DPV because of invalid secondary
93	Failed DPV because of missing secondary

Uncorrected	
111-120	Last Line Parsing
111	No input ZIP, no input state, and no input city
112	No input ZIP and no input city
113	Foreign address
211-220	Last Line Retrieval
211	No input ZIP, no input state, and input city invalid
212	No input ZIP, input state invalid, and input city invalid
213	Input ZIP invalid, no input state, and input city invalid
214	Input ZIP invalid, input state invalid, and input city invalid
215	Input ZIP invalid, no input city
216	Unique input ZIP Code does not match input city/state
311-320	Address Line Parsing
311	Could not parse a primary number from input data
312	Could not parse a street name from input data
313	Address lines blank
411-430	Address Line Retrieval
411	Primary number invalid
412	Street name invalid
413	Similar street names were found but with no exact matches
414	Multiple possible matches with different ZIP+4 codes were found
415	Predirectional required to choose from multiple possible matches
416	Postdirectional required to choose from multiple possible matches
417	Suffix required to choose from multiple possible matches
418	Directional and suffix required to choose from multiple possible matches
419	Valid ZIP Code required to choose from multiple possible matches
420	Valid city name required to choose from multiple possible matches
421	Valid urbanization required to choose from multiple possible matches
422	Matched to undeliverable address; 5-digit coded
423	Matched to a record in the Early Warning System (EWS) file; an exact match will be possible with the next database update
491-499	DPV Failures
491	Failed DPV because of invalid primary
492	Valid primary but failed DPV because of invalid secondary
493	Valid primary but failed DPV because of missing secondary
494	Failed DPV because of invalid PO, RR, or HC box number