



Address Verification, Correction and ZIP+4 API User's Guide

The Software Company, Inc.
<https://SoftwareCompany.com>



NetZipCode for .NET allows you to quickly and easily build Zip Code lookup, address verification plus address correction and parsing into your custom applications. The delivery address is checked for proper Postal Service format and standardized if necessary. Spelling corrections are made and the address is split into USPS standard components.

NetZipCode for .NET not only returns a standardized and corrected address, but also provides the following additional information about each address:

- Individually Parsed Address Components
- Zip+4 Add-On
- Preferred/Default City Name
- County Name
- Congressional District
- Zip Code Latitude/Longitude

An array of diagnostic flags is returned with each address letting you know exactly what strategy was used in order to achieve the best match.

Benefits

- **Save \$\$\$ on Postage** – verify addresses before you ship or mail
- **Catch Data-Input Errors** – virtually eliminating undeliverable addresses
- **Save Keying Time** – city/state is returned from 5-digit Zip Code
- **Unlimited Use** – no escalating charges
- **Free Database Updates** for a full year

Features

- Addresses are standardized to USPS recommended abbreviations
- No automatic database expiration
- Proper case conversion for more attractive data presentation
- Attractive USPS database subscription pricing
- Designed for use with all .NET-compatible programming languages

NetZipCode for .NET follows the US Postal Service “one component failure rule” to find a correct address match. This rule permits an address match if no more than one address component mismatch exists, where a “component” is defined as a predirectional, street name, street suffix, or a postdirectional.

Consider the following address: 100 E MAIN ST N

This address has all four components. If one of the components has to be added, changed, or deleted in order to achieve a unique match, NetZipCode will correct the address and return the Zip+4. Under USPS address matching rules, if more than one component is incorrect, no match is allowed. If adding, changing or deleting a component results in multiple matches, NetZipCode will return a list of all matching addresses.

Example

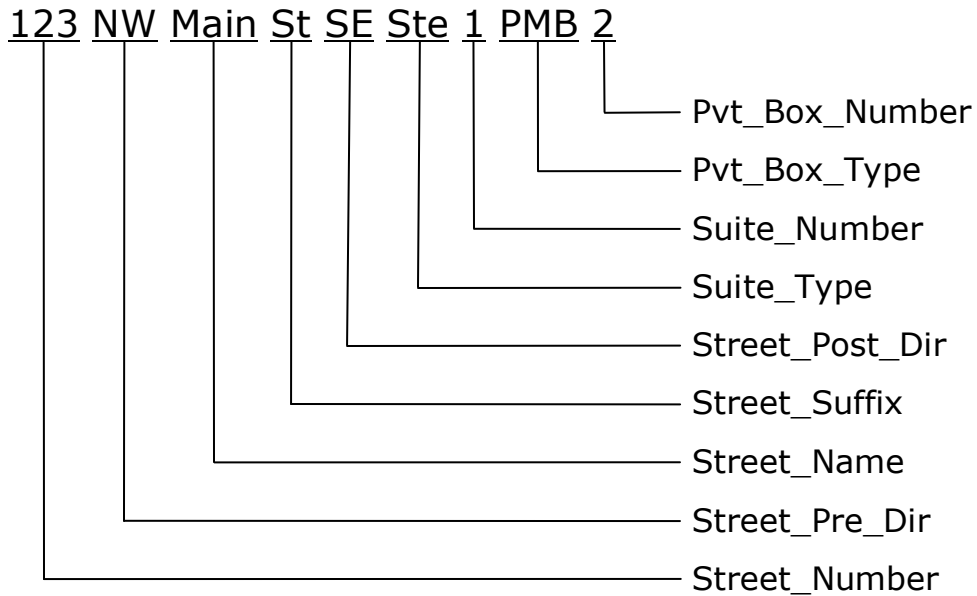
Input Address: 9227 2 3
32824

Output Address: 9227 2nd Ave, Ste 3
Orlando, FL 32824-8390
County: Orange
Preferred City: Orlando

Results:

- Street number/name verified: (deliverable address)
- Street name corrected: (2nd)
- Street suffix added: (Ave)
- Suite type added: (Ste)
- City/state added: (Orlando, FL)
- ZIP+4 added: (8390)
- County name added: (Orange)
- Preferred city added: (Orlando)
- Capitalization applied

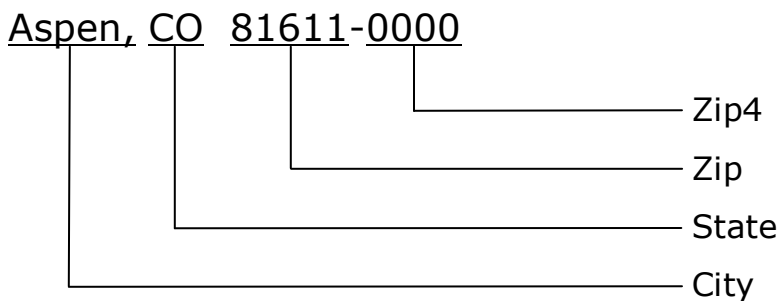
Street Address



Box Address



City/State/Zip



Address_In

Syntax: Address_In = String

Description:

Set this property to the address string to be processed. When the “Lookup” method is invoked, the “Address_In” and “CSZ_In” strings are matched to the database. When a match is found, the standardized and corrected address is placed into the “Address_Out” and “CSZ_Out” properties. In addition, each element of the “Address_In” and “CSZ_In” strings is placed into the appropriate address and city/state/zip component property.

Suite_In (optional – use when suite is in a separate field)

Syntax: Suite_In = String

Description:

Set this property to the suite number string when it’s not already included in the “Address_In” property. When a match is found, the standardized and corrected suite is placed into the “Suite_Type” and “Suite_Number” properties.

CSZ_In

Syntax: CSZ_In = String

Description:

Set this property to the city, state and/or Zip Code string to be processed. When the “Lookup” method is invoked, the “CSZ_In” string is matched to the database. If a match is found, the standardized and corrected city/state/zip is placed into the “CSZ_Out” property. In addition, each element of the “CSZ_In” string is placed into the appropriate city/state/zip component property. *If this property is set to the city and state, the Zip Code is optional. If set to the 5-digit Zip Code, the city and state are optional.*

Firm_Name_In (optional)

Syntax: Firm_Name_In = String

Description:

Set this property to the firm name or high-rise name string to be processed. This will produce the highest level of match for the most precise 4-digit Zip+4 add-on.

PR_Urb_In (optional)

Syntax: PR_Urb_In = String

Description:

Set this property to the Puerto Rican urbanization string to be processed.

Capitalization

Syntax: Capitalization = StringLiteral

Description:

Set this property to “Upper”, “Lower” or “Mixed” to indicate your capitalization preference for the output address and its components. **Default is “Upper”.**

Pvt_Box_Types

Syntax: Pvt_Box_Types = StringArray

Description:

Set this property to a list of private box types you want NetZipCode to recognize. Some common types are: “PMB”, “MAC”, “Mailstop”, etc. *See “Pvt_Box_Type” and “Pvt_Box_Number” properties.*

Database_Path

Syntax: Database_Path = String

Description:

Set this property to the full path to the database file: “NetZipCode.db”. By default, this file is located in the NetZipCode installation folder. However, you can relocate this file to any other folder as long as you set this property to the full path. **Default is first the folder of the invoking application: “AppDomain.CurrentDomain.BaseDirectory” then the NetZipCode installation folder.**

Static_Key_Name (licensed version)

Syntax: Static_Key_Name = String

Description:

Set this property to the name portion of the static key assignment or blank.

Static_Key (licensed version)

Syntax: Static_Key = String

Description:

Set this property to the key portion of the static key assignment or blank.

Address_Out (read only)

Syntax: String = Address_Out

Description:

After invoking the “Lookup” method, this property will contain the standardized and corrected address string from the “Address_In” property including a suite number if present. If no match was found in the USPS database, this property will be blank.

Address_Out_Street (read only)

Syntax: String = Address_Out_Street

Description:

After invoking the “Lookup” method, this property will contain the street portion of “Address_Out”.

Address_Out_Suite (read only)

Syntax: String = Address_Out_Suite

Description:

After invoking the “Lookup” method, this property will contain the suite portion of “Address_Out”.

Street_Number (read only)

Syntax: String = Street_Number

Description:

After invoking the “Lookup” method, this property is set to the primary address number component of “Address_Out” also referred to as house number or primary number.

Street_Pre_Dir (read only)

Syntax: String = Street_Pre_Dir

Description:

After invoking the “Lookup” method, this property is set to the Predirectional component of “Address_Out”. Values will be a valid directional (N, NE, S, SE, etc.) or blank.

Street_Name (read only)

Syntax: String = Street_Name

Description:

After invoking the “Lookup” method, this property is set to the Street Name component of “Address_Out”. Value will be alphanumeric.

Street_Suffix (read only)

Syntax: String = Street_Suffix

Description:

After invoking the “Lookup” method, this property is set to the Street Suffix component of “Address_Out”. Values will be a valid suffix (St, Ave, Rd, etc.) or blank.

Street_Post_Dir (read only)

Syntax: String = Street_Post_Dir

Description:

After invoking the “Lookup” method, this property is set to the Postdirectional component of “Address_Out”. Values will be a valid directional (N, NE, S, SE, etc.) or blank.

Suite_Type (read only)

Syntax: String = Suite_Type

Description:

After invoking the “Lookup” method, this property is set to the Suite Type component of “Address_Out”. Values will be only valid suite types (Apt, Suite, Unit, etc.) or blank.

Suite_Number (read only)

Syntax: String = Suite_Number

Description:

After invoking the “Lookup” method, this property is set to the Suite Number component of “Address_Out”. Values will be alphanumeric suite number or blank.

Box_Type (read only)

Syntax: String = Box_Type

Description:

After invoking the “Lookup” method, this property is set to the Box Type component of “Address_Out”. Values will be only valid box types (PO Box, RR, HC, etc.) or blank.

Box_Type_Number (read only)

Syntax: String = Box_Type_Number

Description:

After invoking the “Lookup” method, this property is set to the Box Type Number component of “Address_Out”. Values will be alphanumeric box type number or blank.

Box (read only)

Syntax: String = Box

Description:

After invoking the “Lookup” method, this property is set to the Box component of “Address_Out”. Value will be “Box” or blank.

Box_Number (read only)

Syntax: String = Box_Number

Description:

After invoking the “Lookup” method, this property is set to the Box Number component of “Address_Out”. Values will be alphanumeric box number or blank.

Pvt_Box_Type (read only)

Syntax: String = Pvt_Box_Type

Description:

After invoking the “Lookup” method, this property is set to the Private Box Type component of “Address_Out”. Values will be valid box types as specified in the “Pvt_Box_Types” property.

Pvt_Box_Number (read only)

Syntax: String = Pvt_Box_Number

Description:

After invoking the “Lookup” method, this property is set to the Private Box Number component of “Address_Out”. Values will be alphanumeric private box number or blank.

CSZ_Out (read only)

Syntax: String = CSZ_Out

Description:

After invoking the “Lookup” method, this property will contain the standardized city/state/zip (last line) from the “CSZ_In” property.

City (read only)

Syntax: String = City

Description:

After invoking the “Lookup” method, this property is set to the City component of “CSZ_Out”. See “*Preferred_City_Name*” property.

State (read only)

Syntax: String = State

Description:

After invoking the “Lookup” method, this property is set to the State component of “CSZ_Out”. Values will be only valid USPS state abbreviations (FL, AZ, CO, etc.) or blank.

Zip (read only)

Syntax: String = Zip

Description:

After invoking the “Lookup” method, this property is set to the 5-digit Zip Code component of “CSZ_Out”. Values will be a 5-digit numeric Zip Code or blank.

Zip4 (read only)

Syntax: String = Zip4

Description:

After invoking the “Lookup” method, this property is set to the 4-digit Zip+4 add-on component of “CSZ_Out”. Values will be a 4-digit numeric Zip+4 add-on code (sector/segment) or blank.

Zip_Latitude (read only)

Syntax: String = Zip_Latitude

Description:

After invoking the “Lookup” method, this property is set to degrees of latitude of the Zip Code centroid for the 5-digit “Zip” property. For military Zip Codes this will be zero.

Zip_Longitude (read only)

Syntax: String = Zip_Longitude

Description:

After invoking the “Lookup” method, this property is set to degrees of longitude of the Zip Code centroid for the 5-digit “Zip” property. For military Zip Codes this will be zero.

Preferred_City (read only)

Syntax: String = Preferred_City

Description:

After invoking the “Lookup” method, this property is set to the preferred/default city name for the 5-digit “Zip” property. See “*Preferred_City_Name*” and “*Preferred_State*” properties.

Preferred_State (read only)

Syntax: String = Preferred_State

Description:

After invoking the “Lookup” method, this property is set to the preferred/default state abbreviation for the 5-digit “Zip” property. Values will be valid USPS state abbreviations (FL, AZ, CO, etc.) or blank. See “*Preferred_City_Name*” and “*Preferred_City*” properties.

Match_Count (read only)

Syntax: Integer = Match_Count

Description:

After invoking the “Lookup” method, this property is set to an integer count of the number of matching addresses when the “Multiple_Match” property is set to “True”. See “*GetFirstMatch*”, “*GetNextMatch*” methods and “*Multiple_Match*” property.

Address_Type (read only)

Syntax: String = Address_Type

Description:

After invoking the “Lookup” method, this property is set to one of the following:

S	Street	(1 N Main St, 2 US Highway 285, etc.)
P	Post Office Box	(PO Box 1)
R	Rural Route	(RR 1 Box 2)
H	Highway Contract	(HC 1 Box 2)
G	General Delivery	(General Delivery, Gen Del, GD, etc.)
M	Military	(CMR 1 Box 2, etc.)
N	Not a valid address	(“Address_Match” property will also be set to “False”)

Zip_Type (read only)

Syntax: String = Zip_Type

Description:

After invoking the “Lookup” method, this property is set to one of the following:

P	PO Box Only	(no street addresses in this Zip Code)
M	Military Zip	(military/embassy specific Zip Code - APO/FPO/DPO)
U	Unique Zip	(assigned to a single business or organization)
S	Standard Zip	(any combination of street addresses and PO Boxes)
N	Not a valid Zip	(“CSZ_Match” property will also be set to “False”)

Alias_Type (read only)

Syntax: String = Alias_Type

Description:

After invoking the “Lookup” method, when the input address is matched to a street alias, this property is set to the type of alias match. *See “Alias_Match” property.*

A	Abbreviated	(abbreviated street name)
C	Changed	(street name changed)
O	Other/nickname	(other/nickname)
P	Preferred	(preferred/default street name)
N	No alias match	(street name was not matched to a street alias)

County (read only)

Syntax: String = County

Description:

After invoking the “Lookup” method, this property is set to the name of the county in which the address is located. Values will be alphanumeric county name or blank.

County_FIPS (read only)

Syntax: String = County_FIPS

Description:

After invoking the “Lookup” method, this property is set to the Federal Information Processing Standard (FIPS) code for the county in which the address resides. Values will be a three-digit numeric string or blank.

State_FIPS (read only)

Syntax: String = State_FIPS

Description:

After invoking the “Lookup” method, this property is set to the Federal Information Processing Standard (FIPS) code for the state in which the address resides. Values will be a two-digit numeric string or blank.

DPBC (read only)

Syntax: String = DPBC

Description:

After invoking the “Lookup” method, this property is set to the 12-digit Delivery Point Bar Code string for this Zip Code. Values will be numeric DPBC or blank.

Congressional_District (read only)

Syntax: String = Congressional_District

Description:

After invoking the “Lookup” method, this property is set to the congressional district code for this address. Values will be alphanumeric congressional district or blank.

Firm_Name_Out (read only)

Syntax: String = Firm_Name_Out

Description:

After invoking the “Lookup” method, this property is set to the firm name or high-rise building name from the matching address record. When the “Firm_Name_In” property is specified and there is no match, “Firm_Name_In” will be passed directly to this property. Values will be alphanumeric or blank.

PR_Urb_Out (read only)

Syntax: String = PR_Urb_Out

Description:

After invoking the “Lookup” method, this property is set to the Puerto Rican Urbanization name from optional the “PR_Urb_In” string. Values will be alphanumeric or blank.

Address Flags (read only)

Description:

After invoking the “Lookup” method, the following Boolean (True/False) properties are returned indicating the status of the address lookup.

Syntax:

Address_Match = True (CSZ_Match is also set to “True”)

A match was found in the database for the “Address_In” and “CSZ_In” properties. This condition also sets one or more of the following properties indicating any corrections that were made in order to achieve a match:

Address_Corrected = True

One or more address components were corrected in order to achieve a match.

Street_Number_Corrected = True

“Street_Number” component was corrected in order to achieve a match.

Street_Pre_Dir_Corrected = True

“Street_Pre_Dir” component was corrected in order to achieve a match.

Street_Name_Corrected = True

“Street_Name” component was corrected in order to achieve a match.

Street_Suffix_Corrected = True

“Street_Suffix” component was corrected in order to achieve a match.

Street_Post_Dir_Corrected = True

“Street_Post_Dir” component was corrected in order to achieve a match.

Suite_Corrected = True

“Suite_Type” component was corrected in order to achieve a match.

Suite_Unknown = True

“Suite_Number” component was not matched in the database.

POB_Corrected = True

“Box_Type” or “Box_Number” components were corrected in order to achieve a match.

PR_Urb_Corrected = True

Puerto Rican urbanization “PR_Urb_In” component was corrected in order to achieve a match.

Alias_Match = True

Input address was matched to street alias name. *See “Alias_Type” property.*

Address_Match = False

A unique match could not be found in the database for the “Address_In” and “CSZ_In” properties. This condition also sets one or more of the following properties indicating the reason a unique match could not be found:

Multiple_Match = True

More than one address was matched in the database.

See “*GetFirstMatch*”, “*GetNextMatch*” methods and “*Match_Count*” property.

Street_Number_Unknown = True

The “Street_Number” was out-of-range for this street in the database.

Street_Name_Unknown = True

The “Street_Name” was not matched in the database.

CSZ_Unknown = True

The City & State or Zip Code was not matched in the database.

Insufficient_Address_Data = True

There was not enough data to find a match in the database.

CSZ_Match = True

A match was found in the database for the “CSZ_In” property. This condition also sets one or more of the following properties indicating any corrections that were made:

City_Corrected = True

City name component was added or corrected in order to achieve a match.

State_Corrected = True

State component was added or corrected in order to achieve a match.

Zip_Corrected = True

ZIP Code component was added or corrected in order to achieve a match.

Zip4_Corrected = True

ZIP+4 component was added or corrected in order to achieve a match.

CSZ_Match = False (CSZ_Unknown is also set to “True”)

No match was found in the database for the “CSZ_In” property.

No_USPS_Delivery = True

Address was verified, but USPS doesn't deliver to this address.

Preferred_City_Name = True

The "City" property returned is the USPS preferred/default city name for the 5-digit Zip Code. See "*Preferred_City*" and "*Preferred_State*" properties.

Highrise = True

Address is located in a commercial building, apartment complex, high-rise, wing or floor of a building, grouping of apartment mail boxes or other physical location other than a street.

Firm = True

The "Firm" flag is set to "True" when the "Firm_Name_In" property was matched to an address record. This is the finest level of match and will, in some cases, return a different Zip+4 add-on code as a result of the match.

Cross_State_Match = True

The "Cross_State_Match" flag is set to "True" when the physical address is located in a different city/state than the city/state/zip specified in the "CSZ_In" property. The city and state output properties are corrected to the actual Post Office that serves this address.

Return_Code (read only)

Syntax: String = Return_Code

Description:

After invoking the "Lookup" method, this property is set to blank upon successful completion. Most exceptions occur on the first invocation. The most common ones are listed below. *This property should be examined on each return from NetZipCode.*

Common Return Codes:

Z01 – Z16	"NetZipCode.db" database initialization failed
Z17	"NetZipCode.db" database version, "NetZipCode.dll" version mismatch
Z35	"NetZipCode.db" database file not found (<i>see "Database_Path" property</i>)
L00	Evaluation period expired
L01	Static key validation failed (<i>see "Static_Key" property</i>)
L50	Evaluation license error

Clear

Syntax: NetZipCode.Clear

Description:

When this method is invoked, all properties are cleared with the exception of “Static_Key”, “Static_Key_Name” and “Database_Path”.

Lookup

Syntax: NetZipCode.Lookup

Description:

When this method is invoked, the “Address_In” and “CSZ_In” properties are compared to the USPS National Address Database. When a match is found, the address is standardized and corrected then placed into the “Address_Out” and “CSZ_Out” properties. In addition, each standardized element of the “Address_Out” and “CSZ_Out” properties is placed into the appropriate address component property. Next, the Address Flags are set to indicate the status of the address lookup as well as how complete and correct the address is. The “Return_Code” property is also set and should be checked after each invocation of the “Lookup” method. *See “Return_Code” property.*

GetFirstMatch

Syntax: NetZipCode.GetFirstMatch

Description:

This method can only be invoked when the “Multiple_Match” flag is set to “True”. After invoking the “GetFirstMatch” method, the first address from the current group of matching addresses is returned and the “Address_Match” flag is set to “True”. *See “Multiple_Match” and “Match_Count” properties.*

GetNextMatch

Syntax: NetZipCode.GetNextMatch

Description:

This method can only be invoked *after* the “GetFirstMatch” method has been invoked and when the “Multiple_Match” flag is set to “True”. After invoking the “GetNextMatch” method, the next address from the current group of matching addresses is returned and the “Address_Match” flag is set to “True”. When no more addresses remain, the “Address_Match” property is set to “False”. *See “Multiple_Match” and “Match_Count” properties.*

GetFirstCityInZip

Syntax: NetZipCode.GetFirstCityInZip

Description:

When this method is invoked, the 5-digit “CSZ_In” property is compared to the USPS National Address Database. When a match is found, the first city name (alphabetically) in the list of USPS-approved mailing names for this Zip Code is returned in the “City” property and the “CSZ_Match” property is set to “True”. If this is the USPS preferred/default city name for this Zip Code, then the “Preferred_City_Name” property is set to “True” and the “Preferred_City” property is set to the city name. See “Preferred_City_Name”, “Preferred_City” and “Preferred_State” properties.

GetNextCityInZip

Syntax: NetZipCode.GetNextCityInZip

Description:

This method can only be invoked *after* the “GetFirstCityInZip” method has been invoked and a match was found (CSZ_Match = True). After invoking the “GetNextCityInZip” method, the next city name (alphabetically) in the list of USPS-approved mailing names for this Zip Code is returned in the City property and the “CSZ_Match” property is set to “True”. If this is the USPS preferred/default city name for this Zip Code, then the “Preferred_City_Name” property is set to “True” and the “Preferred_City” property is set to the city name. See “Preferred_City_Name”, “Preferred_City” and “Preferred_State” properties.

DistanceBetweenZips

Syntax: String = NetZipCode.DistanceBetweenZips(Zip1, Zip2)

Description:

When this method is invoked, a calculation is made of the distance between “Zip1” and “Zip2” centroids and the distance in miles is returned (+/- 36 feet). If either “Zip1” or “Zip2” is a military Zip Code or a match cannot be found in the database, this method returns a value of zero and the “CSZ_Unknown” flag is set to “True”.

ZipsInRadius

Syntax: StringArray = NetZipCode.ZipsInRadius(Zip, RadiusMiles)

Description:

When this method is invoked, all Zip Codes within the “RadiusMiles” distance (+/- 36 feet) from the “Zip” centroid are returned in a 2-dimensional string array (Zip, Miles). If “Zip” is a military Zip Code or a match cannot be found in the database or no Zips can be found within “RadiusMiles”, this method returns an empty array and the “CSZ_Unknown” flag is set to “True”.

SeasonalDelivery

Syntax: Boolean = NetZipCode.SeasonalDelivery(Zip, Month)

Description:

When this method is invoked, a Boolean (True/False) value is returned indicating whether or not mail is delivered to the specified “Zip” in the specified “Month”. This property will be set to “True” if mail delivery is provided in the specified “Month”. When “Month” is zero, this will be set to “True” only when mail is delivered in all 12 months.

Deploying Your Applications

Be sure to include the following in your deployment package:

NetZipCode.dll – usually placed in the application folder or Global Assembly Cache (GAC)

NetZipCode.db – usually placed in the application folder*

* “NetZipCode.db” database file can be placed anywhere on the target machine as long as the full path to it is specified in the “Database_Path” property.

In addition to the above, there is a common runtime that can be placed in the application folder or the Global Assembly Cache (GAC) of the target machine.

Fujitsu.COBOL.dll

Evaluation License

The evaluation license is valid for a period of 7 days or up to 1,000 calls.

Sales@SoftwareCompany.com

Support@SoftwareCompany.com