

**Active**  
**Address** activex  
**4.0**

**Address Validation & Standardization Component**

The Software Company, Inc.  
[www.SoftwareCompany.com](http://www.SoftwareCompany.com)

**The Software** Software is our middle name  
*Company*™

**ActiveAddress COM Object** allows you to quickly and easily build address verification, standardization and parsing into your custom applications. Accept addresses free-form and let ActiveAddress do the rest. Each address will be standardized, split into USPS standard components then graded for overall completeness and accuracy. Plus, no USPS database subscription is required.

ActiveAddress can process all types of addresses including Suite Numbers and City/State/Zip. A special Address\_Quality flag is returned each time an address is processed allowing you to easily identify questionable addresses before they enter your system. Also returned is the Address\_Type flag indicating the type of address being processed: Street, Military, PO Box, Rural Route, Highway Contract, General Delivery or Suite giving you flexibility in their handling. And, for a more appealing presentation, let ActiveAddress set the proper capitalization.

As a bonus, when you combine ActiveAddress with our ActiveGender product, you can handle even the impossible task of identifying data that has been entered free-form, where the names, addresses and C/S/Z “float” from field to field. You’ll always be certain of what data you’re working with.

ActiveAddress is being used by government agencies from coast-to-coast. Some states now require that addresses be stored in standardized and parsed format to facilitate address matching from agency to agency.

### Benefits

- Save \$\$\$ on Postage by Eliminating Incomplete or Non-Addresses
- Catch Data-Input Errors Before They Enter Your Database
- Standardize Addresses for Faster Processing
- Unlimited Processing Volume - No Recurring Update Charges
- FREE Upgrades for a Full Year

### Features

- Addresses are Standardized to USPS Recommended Abbreviations
- Proper Case Conversion for More Attractive Data Presentation
- Royalty-Free Runtime
- Easily Separate Name, Street Address and City/State/Zip
- Designed for Use with Visual Basic, C++, Visual FoxPro, ASP, Access, SQL and more

**ActiveAddress COM Object** starts by carefully identifying each individual address component based on its context. Intuitive algorithms examine the results and a selection is made of the most complete and correct data. If needed, format corrections are made and the USPS recommended abbreviations are applied. The `Address_Quality` flag is then set to indicate how complete and correct the address is. Finally, the standardized address components are returned to your application along with a complete and cleansed composite address.

Strict conformity to USPS "[Postal Addressing Standards Publication 28](#)" ensures consistent standardization of every address. However, you can easily customize these settings for critical applications.

ActiveAddress COM Object is the *only* address verification and parsing software that can reliably find and extract a Street Address when it's surrounded by extraneous data. It can even separate Street Address from City/State/Zip when they're in the same field.

*The success or failure of any parsing software is dependent on how well it can handle "dirty" addresses. These are addresses that have non-standard abbreviations or the address elements are run together such as APT6. ActiveAddress can handle these and more.*

### Examples

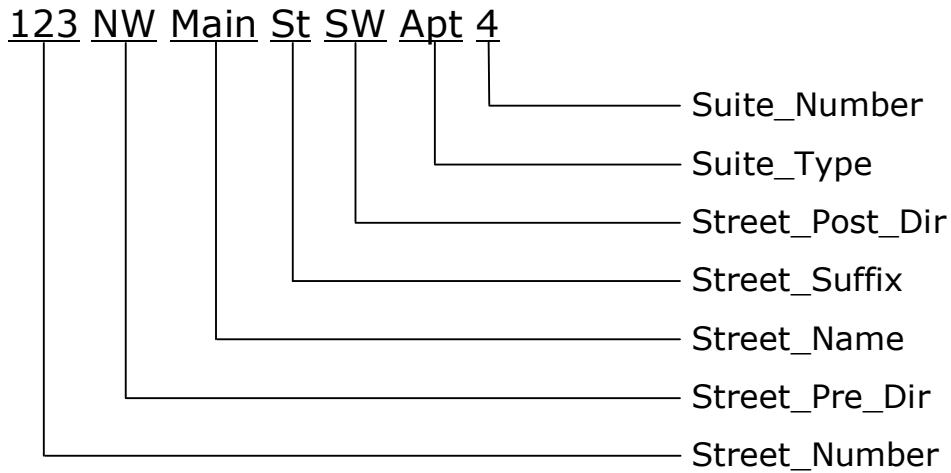
**Address\_In:** IS DEPT 123NE MAIN ST#1  
**Address\_Out:** 123 NE Main St # 1

**Address\_In:** Re: DOC#222 123 ADAMS BL AP5%DORIS  
**Address\_Out:** 123 Adams Blvd Apt 5

**Address\_In:** AP2,123NE SOUTH STREET WEST%JANE  
**Address\_Out:** 123 NE South St W Apt 2

**Address\_In:** 6TH FLOOR ONE BROADWAY ST,RE:LN-123456  
**Address\_Out:** 1 Broadway St Fl 6

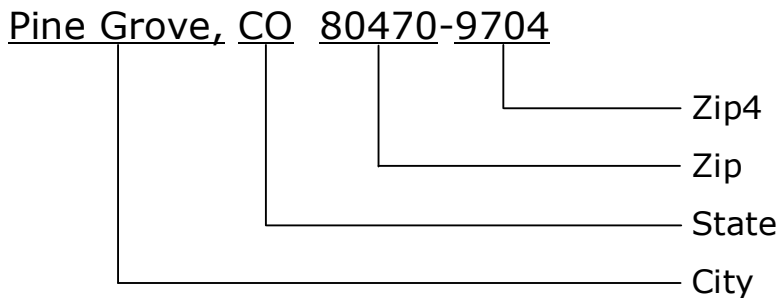
**Street Address**



**Box Address**



**City/State/Zip**



### Address\_In

**Syntax:** Address\_In = String

**Description:**

Set this property to the address string to be processed.

When the “Parse” method is called, the Address\_In string is standardized and corrected then placed into the Address\_Out property. In addition, each element of the Address\_In string is placed into the appropriate address component property.

---

### CSZ\_In

**Syntax:** CSZ\_In = String

**Description:**

Set this property to the city, state and zip string (last line) of the address.

When the “Parse” method is called, the CSZ\_In string is standardized and corrected then placed into the CSZ\_Out property. In addition, each component of the CSZ\_In property is placed into the appropriate City, State, Zip component property.

*Setting CSZ\_In to “USA” or “Canada” will force the appropriate interpretation of Address\_In.*

**Canadian Addresses:**

Municipality / Province / Postcode are synonymous with City / State / Zip respectively.

---

**Address\_CSZ\_Combined** (new in v4.0+)  
*(replaces obsolete “CSZ\_InSameField” property)*

**Syntax:** Address\_CSZ\_Combined = Boolean

**Description:**

Set this property to Boolean (True/False) to indicate whether or not the Address\_In property also contains city/state/zip data. Set Address\_CSZ\_Combined to “True” to parse this extra data into the City, State and Zip properties. **Default is “False”.**

---

**Address\_CSZ\_Delimiter** (new in v4.0+)

**Syntax:** Address\_CSZ\_Delimiter = String

**Description:**

Set this property to an *optional* delimiter string to indicate where to separate the address from the CSZ when Address\_CSZ\_Combined is set to “True”. **Default is no delimiter.**

### **Street\_Number\_Suite\_Combined** (new in v4.0+)

**Syntax:** Street\_Number\_Suite\_Combined = Boolean

**Description:**

Set this property to Boolean (True/False) to indicate whether or not the Street\_Number property also contains a Suite\_Number. Set Street\_Number\_Suite\_Combined to “True” if you want hyphenated street numbers parsed into Street\_Number and Suite\_Number. **Default is “False”.**

USA: 10-100 Main Street = 10 Main St #100

CANADA: 10-100 Main Street = 100 Main St #10

**Note:**

If a suite number is present in the Address\_In string, this setting will be overridden.

---

### **Numeric\_Street\_Conv**

**Syntax:** Numeric\_Street\_Conv = Boolean

**Description:**

Set this property to Boolean (True/False) to indicate whether or not to convert a spelled out ordinal street name to an ordinal number. (“Third” converts to “3<sup>rd</sup>”, etc.) Set Numeric\_Street\_Conv to “True” if you want the street name converted. Numeric street names are spelled out only when there are duplicate street names within a postal delivery area and the only distinguishing factor is that one of them is spelled out. **Default is “False”.**

---

### **Numeric\_Street\_No\_Ordinal** (new in v4.0+)

**Syntax:** Numeric\_Street\_No\_Ordinal = Boolean

**Description:**

Set this property to Boolean (True/False) to indicate whether or not to convert a numeric street name to an ordinal number. (“3” converts to “3<sup>rd</sup>”, etc.) Set Numeric\_Street\_No\_Ordinal to “True” if you want to retain the original numeric street name. **Default is “False”.**

---

### **Output\_Case** (changed in v3.0+)

**Syntax:** Output\_Case = “StringLiteral”

**Description:**

Set this property to “Upper”, “Mixed” or “None” to indicate your capitalization preference for the output address and its components. **Default is “None”.**

### Reference\_File\_Path (changed in v3.0+)

**Syntax:** Reference\_File\_Path = String

**Description:**

Set this property to the full path and file name of the user-defined file containing the Street\_Suffix and Suite\_Type abbreviations. A standard set of abbreviations is supplied and installed in the “Windows\System32” folder under the name: “ActiveAddress.ref”. You can rename and relocate this file to any other folder as long as you set the Reference\_File\_Path property to the full path and file name. **Default is “Windows\System32” folder.**

*See “Updating User Control Tables” later in this guide for instructions on customizing this file.*

---

### Static\_Key\_Name

**Syntax:** Static\_Key\_Name = String

**Description:**

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

---

### Static\_Key

**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 calling the “Parse” method, this property will contain the standardized and corrected address string from the Address\_In property including a suite number if present. If the Address\_In string returns an Address\_Quality of “Low”, this property will be blank.

---

### **Address\_Out\_Street** (read only)

**Syntax:** String = Address\_Out\_Street

**Description:**

After calling the “Parse” method, this property will contain the street portion of Address\_Out.

---

### **Address\_Out\_Suite** (read only)

**Syntax:** String = Address\_Out\_Suite

**Description:**

After calling the “Parse” method, this property will contain the suite portion of Address\_Out.

---

### **Street\_Number** (read only)

**Syntax:** String = Street\_Number

**Description:**

After calling the “Parse” method, this property is set to the primary address number component of Address\_Out commonly referred to as house number, street number, civic number or range.

---

### **Street\_Pre\_Dir** (read only)

**Syntax:** String = Street\_Pre\_Dir

**Description:**

After calling the “Parse” 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 calling the “Parse” 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 calling the “Parse” 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 calling the “Parse” 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 calling the “Parse” 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 calling the “Parse” 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 calling the “Parse” 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 calling the “Parse” 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 calling the “Parse” method, this property is set to the Box component of Address\_Out. Value will be “Box” or blank.

**Canadian Addresses:**

Value may also be “Stn” and “RPO”

---

### **Box\_Number** (read only)

**Syntax:** String = Box\_Number

**Description:**

After calling the “Parse” method, this property is set to the Box Number component of Address\_Out. Values will be alphanumeric box number or blank.

**Canadian Addresses:**

Value may also be Station name or Retail Postal Outlet (RPO) name.

---

### **Address\_Quality** (read only)

**Syntax:** String = Address\_Quality

**Description:**

After calling the “Parse” method, this property is set according to the completeness of the Input\_Address. "Low" if no recognizable address is present. "Medium" if an address is present but is incomplete, such as a missing apartment number or street suffix. "High" is returned when a complete and technically correct address or suite is found.

If your addresses “float” from field to field you can easily determine which field contains the address through trial and error by examining Address\_Quality and Address\_Type after trying each field.

### **Address\_Type** (read only)

**Syntax:** String = Address\_Type

**Description:**

After calling the “Parse” method, this property is set to one of the following address types:

<b>S</b>	Street	(1 N Main St, 2 US Highway 285, etc.)
<b>A</b>	Suite	(Apt 1, Suite 2, etc.)
<b>P</b>	Post Office Box	(PO Box 1)
<b>R</b>	Rural Route	(RR 1 Box 2)
<b>H</b>	Highway Contract	(HC 1 Box 2)
<b>G</b>	General Delivery	(General Delivery, Gen Del, GD, etc.)
<b>M</b>	Military	(CMR 1 Box 2, etc.)
<b>N</b>	Not a valid address	Address_Quality will also be set to “Low”

---

### **Address\_Leading\_Data** (read only)

**Syntax:** String = Address\_Leading\_Data

**Description:**

After calling the “Parse” method, this property will contain all data that precedes the actual address. If the Address\_In string returns an Address\_Quality of “Low”, this property will be blank.

---

### **Address\_Trailing\_Data** (read only)

**Syntax:** String = Address\_Trailing\_Data

**Description:**

After calling the “Parse” method, this property will contain all data that follows the actual address. If the Address\_In string returns an Address\_Quality of “Low”, this property will be blank.

---

### **Address\_Filtered\_Data** (read only)

**Syntax:** String = Address\_Filtered\_Data

**Description:**

After calling the “Parse” method, this property will contain all data that was filtered out before processing according to the [FilterAddress] section of ActiveAddress.ref file.

### **CSZ\_Out** (read only)

**Syntax:** String = CSZ\_Out

**Description:**

After calling the “Parse” method, this property will contain the standardized city/state/zip (last line) from the CSZ\_In property.

**Canadian Addresses:**

Municipality / Province / Postcode are synonymous with City / State / Zip respectively.

---

### **City** (read only)

**Syntax:** String = City

**Description:**

After calling the “Parse” method, this property is set to the City component of CSZ\_Out.

**Canadian Addresses:**

Municipality will be placed in City.

---

### **State** (read only)

**Syntax:** String = State

**Description:**

After calling the “Parse” 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.

**Canadian Addresses:**

Province abbreviation will be placed in State.

---

### **Zip** (read only)

**Syntax:** String = Zip

**Description:**

After calling the “Parse” method, this property is set to the Zip component of CSZ\_Out. Values will be a 5-digit numeric zip code or blank.

**Canadian Addresses:**

Forward Sortation Area will be placed in Zip. This is the left segment of the postcode: A1A 1A1

### **Zip4** (read only)

**Syntax:** String = Zip4

**Description:**

After calling the “Parse” method, this property is set to the Zip+4 component of CSZ\_Out. Values will be a 4-digit numeric zip add-on code (sector/segment) or blank.

**Canadian Addresses:**

Local Delivery Unit will be placed in Zip4. This is the right segment of the postcode: A1A 1A1

---

### **Country** (read only) (changed in v4.0+)

**Syntax:** String = Country

**Description:**

After calling the “Parse” method, this property will contain the country identified by the input state/province. Values will be the ISO 3166 standardized country code: “US” (USA), “CA” (Canada) or blank.

---

### **State\_FIPS** (read only) (new in v4.0+)

**Syntax:** String = State\_FIPS

**Description:**

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

---

### **CSZ\_Quality** (read only)

**Syntax:** String = CSZ\_Quality

**Description:**

After calling the “Parse” method, this property is set to “Low”, “Medium” or “High” to indicate the probability that the CSZ\_In property value is complete and correct.

---

### **CSZ\_Filtered\_Data** (read only) (new in v4.0+)

**Syntax:** String = CSZ\_Filtered\_Data

**Description:**

After calling the “Parse” method, this property will contain all data that was filtered out before processing according to the [FilterCityStateZip] section of ActiveAddress.ref file.

### **Return\_Code** (read only)

**Syntax:** String = Return\_Code

#### **Description:**

After calling the “Parse” method, this property is set to blank upon successful completion. Some common exceptions are listed below. Most exceptions usually occur on the first call to the “Parse” method. *This property should be examined on each return from ActiveAddress.*

#### **Common Return Codes:**

<b>R35</b>	Reference File Not Found (see Reference_File_Path property)
<b>T00</b>	Suffix Table Limit Reached (1024)
<b>T01</b>	Suite Type Table Limit Reached (256)
<b>T02</b>	Filter Table Limit Reached (128)
<b>L00</b>	Evaluation Period Expired
<b>L01</b>	Static Key Validation Failed (see Static_Key property)
<b>L50 - L69</b>	License Validation Failed

### Clear

**Syntax:**       ActiveAddress.Clear

**Description:**

When this method is called, all properties are set to null with the exception of Static\_Key, Static\_Key\_Name and Reference\_File\_Path.

---

### Parse

**Syntax:**       ActiveAddress.Parse

**Description:**

When this method is called, the Address\_In property is standardized and corrected then placed into the Address\_Out property. In addition, each element of the Address\_Out property is placed into the appropriate address component property and the Address\_Quality and Address\_Type flags are set. The Return\_Code property is also set and should be checked after each call to the "Parse" method. *See "Return\_Code" property.*

### Updating User Control Tables

**ActiveAddress.ref** contains a complete list of street suffixes and suite types along with their abbreviations and full spellings. It is located in the “Windows\System32” folder. Use Notepad or a similar text editor to edit the file. Detailed instructions on the format of the entries are contained within the file. This file can also be relocated. *See Reference\_File\_Path property.*

ActiveAddress allows you to specify which street suffixes and suite types are to be recognized as well as your preferred abbreviations.

The filter section of the table allows you to specify which, if any, characters, words or phrases are to be ignored during processing. All filters that were found in the Address\_In property will be stored in the Address\_Filtered\_Data property.

- 1st Column: Common
- 2nd Column: Full Spelling
- 3rd Column: Abbreviation
- 4<sup>th</sup> Column: City Prefix Flag (Y/N) - When Address\_CSZ\_Combined is set to "True" and a street suffix is already present, treat this suffix as part of the city name.

#### [StreetSuffixUSA]

ST	Street	St	Y
STR	Street	St	N
STREET	Street	St	N

#### [SuiteType]

STE	Suite	Ste	Y
SUITE	Suite	Ste	Y

#### [FilterAddress]

C/O  
ET AL

*Note: Changing the “Full Spelling” column (column 2) is not recommended.*

### If Installation Doesn't Start Automatically:

- Select **Start > Run** from the Task Bar.
- Type CD-ROM drive letter followed by “:\ActiveAddress40.msi” and press enter.

In the folder “Program Files\The Software Company\ActiveAddress 4.0” you will find a sample program named: Sample.vbp. There is also a compiled version called Sample.exe that you can run to demonstrate ActiveAddress.

---

### Deploying Your Applications

“ActiveAddress.dll” is usually placed in the application folder of the target machine and must be registered with Windows. You can use “Regsvr32.exe” for this purpose. The “Car\*.dll” files are usually placed in the “Windows\System32” folder. The “ActiveAddress.ref” file is located, by default, in the “Windows\System32” folder but can be relocated anywhere on the target machine as long as the full path and file name are specified in the Reference\_File\_Path property.

Be sure to include all of the following in your deployment package:

**ActiveAddress.dll** – register with Windows (you can use Regsvr32.exe for this purpose)

In addition to the above, there is a small runtime package which is usually installed into “Windows\System32 folder”. *These do not need to be registered with Windows.*

**Carcla40.dll**  
**Carclw60.dll**  
**Carfsw20.dll**  
**ActiveAddress.ref\***

\* ActiveAddress.ref can be relocated anywhere on the target machine as long as the full path and file name are specified in the Reference\_File\_Path property.

---

This product is initially licensed for a period of 30 days or up to 1000 calls. It must be registered to continue using it after this evaluation period. Please contact us at:

[Sales@SoftwareCompany.com](mailto:Sales@SoftwareCompany.com)

303/838-1223 (voice)

303/838-1224 (fax)