



OUR TECHNOLOGY. YOUR SUCCESS.

# AccuZIPToolkit

**Data Quality Developer Toolkit**



### **AccuZIP Data Quality Developer Toolkit**

WARNING: This product contains a “disabling clock” which disables all functionality of the Toolkit if the database is no longer current due to Postal Service requirements. You must purchase updates for the upcoming period in order to continue utilizing the AccuZIP Data Quality Toolkit.

### **Calling the AccuZIP Data Quality Toolkit from Other Applications**

The Toolkit provides an interface that allows programmers to embed the AccuZIP Data Quality Toolkit within their applications.

The Developer’s Toolkit enables applications to make calls to the Functions within the Toolkit to Standardize Addresses. The input data (Company, Address, Address2, City, State, ZIP Code, and Urbanization) needs to be in a specific format described below. The Standardized output format is also described in detail below.

The AccuZIP Data Quality Developer Toolkit works with the dBase programming language, including other languages, such as Visual Basic, C#, C++, C, FoxPro and Java. The Toolkit is transparent when the application that is using it is running.



# Getting Started

## Step 1

1. Run the Setup32.exe (32bit DLL) or Setup64.exe (64bit DLL) installer, whichever matches your project environment.
2. Extract and move the “ADRDATA” folder from the USPSDatabaseYYYYMMDD.zip archive to the “Destination Folder” location selected when you ran the Setup32.exe or Setup64.exe in Step 1.

Note: If you later move the AccuAddress.dll from its original location, please use the .NET REGASM.EXE to Unregister the AccuAddress.dll from the original location before Registering the AccuAddress.dll from the new location. Always move the entire folder content to your project location.

### Program Files

#### AccuAddressDll\_32bit and/or AccuAddressDll\_32bit folder content

- AccuAddressUnregister32bit.bat or AccuAddressUnregister32bit.bat
  - Use to Unregister the AccuAddress.dll when Run as Administrator from the same folder location as the AccuAddress.dll.
- AccuZIPToolkit.pdf
  - Toolkit Documentation
- log4net.dll
- log4net.xml
  - Used for logging AccuAddress.dll
  - Add LOGGINGENABLED=1 to turn ON logging. Not recommended for Production environment.
  - LOGGINGENABLED=0 or omitted to turn logging OFF.
- Support
  - addrcode.dbf
    - Required file and referenced in Config.acu
  - Config.acu
    - Contains the full paths to the USPS Database files. Please ensure these are correct before beginning instantiating the object.
- Toolkit Samples
  - CASSCertificationWebService.zip
  - CSharp.zip
  - JAVAtoCASSrestAPI.zip
  - Unmanaged.zip
  - VisualBasic.zip
  - WebServiceAddressValidator.NET.zip
- AccuAddress.dll
- AccuAddress.dll.config
- AccuAddress.pdb
- AccuAddress.tlb
  - AccuAddress toolkit and dependencies
- AccuAddressRegister32bit.bat
  - Use to Register the AccuAddress.dll when Run as Administrator from the same folder location as the AccuAddress.dll.



Note: Canada references below are used only if you purchased the Canadian Address Verification separately.

```
[CONFIG]
NATIONAL=PathTo\National\
LOT= PathTo\National\
EWS= PathTo\National\
DPV= PathTo\dpv\
LACSLINK= PathTo\LACSLink\
SUITELINK= PathTo\SuiteLink\
CANADADATADIR=PathTo\Canada\
LOGGINGENABLED=0

[SYSTEM]
OUTPUT= PathTo\Support\AddrCode.dbf
CANADADLL=PathTo\PCLookup.dll
```

**Database files:**

Database folder

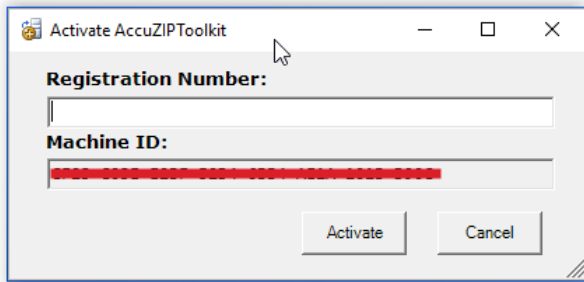
- Canada
- DPV
- LACSLink
- Lookup
- NATIONAL
- SUITELink

**IMPORTANT:** The Database\National\ folder contains a file named "Plus4cfg.dat". This file must remain in this folder and its Permissions must be set to Full Control with Modify, Read, Write checked.



# Step 2

When `zp4connect(path,msg)` or `zp4connectalt(path)` is called for the first time, or if the product is Expired, after you remove the `\Support\Address.dat` file, an “Activate AccuZIPToolkit” dialog box will appear. Send the Machine ID: to [sales@accuzip.com](mailto:sales@accuzip.com). You will receive a Registration Serial Number that can be entered into the “Registration Number” box to activate the dll.



## Calling the AccuZIPToolkit from C#

Step 1: Add ACCUCASS Class into your project --> ACCUCASS.cs

Step 2: Initialized AccuCass class

```
AccuCass Cass = AccuCass();
```

Step 3: Open zp4

```
Cass.Init();
```

Step 4: Lookup Address

```
Cass.Clear();  
Cass.inputCompanyName = CompanyNameVariable.Text.ToString();  
Cass.inputPrimaryAddress = Address.Text.ToString();  
Cass.inputSecondaryAddress = Address2Txt.Text.ToString();  
Cass.inputCityName = citytxt.Text.ToString();  
Cass.inputStateName = statetxt.Text.ToString();  
Cass.inputZipCode = zipcodeTxt.Text.ToString();  
Cass.inputUrbanization = UrbanTxt.Text.ToString();  
Cass.CASSLookup();
```

Step 5: Get Standardized Address

CASS. will have list of output variable available (i.e Cass.resultCompanyFirm,..)

Step 6: Close zp4 Engine

```
if (Cass.Initialized == 1) Cass.Close()
```

Note: You should call `Cass.Init()` and `Cass.Close()` only once in a session and not for each record.

You can test to make sure everything is installed properly by launching `CSharp\AccuCASSCSharpSample\bin\Release\AccuCASSCSharpSample.exe`, then click Address Lookup



# Calling the AccuZIPToolkit from Visual Basic Sample

Step 1: Unzip the \Toolkit Samples\CSharp.zip

Step2: Add ACCUCASS Class into your project --> ACCUCASS.VB

Step 3: Initialized AccuCass class

```
AccuCass Cass = AccuCass()
```

Step 3: Open zp4

```
Cass.Init()
```

Step 4: Lookup Address

```
Cass.Clear()
```

```
Cass.inputCompanyName = CompanyNameVariable.Text.ToString()
```

```
Cass.inputPrimaryAddress = Address.Text.ToString()
```

```
Cass.inputSecondaryAddress = Address2Txt.Text.ToString()
```

```
Cass.inputCityName = citytxt.Text.ToString()
```

```
Cass.inputStateName = statetxt.Text.ToString()
```

```
Cass.inputZipCode = zipcodeTxt.Text.ToString()
```

```
Cass.inputUrbanization = UrbanTxt.Text.ToString()
```

```
Cass.CASSLookup()
```

Step 5: Get Standardized Address

[cass.???](#) will have list of variables available (i.e [Cass.resultCompanyFirm](#),... )

Step 6: Close zp4 Engine

```
if (Cass.Initialized == 1) Cass.Close()
```

Note: You should call `Cass.Init()` and `Cass.Close()` only once in a session and not for each record.

You can test to make sure everything is installed properly by launching

C:\AZ6\AccuAddress\VisualBasic\AccuCASSVBSample\bin\Release\AccuCASSVBSample.exe, then click Address Lookup



# Calling the AccuZIPToolkit from C/C++ (UnManaged Interface)

Prerequisite: Microsoft Visual C++ 2005 Redistributable Package (x86) -

<http://www.microsoft.com/download/en/confirmation.aspx?id=3387>

AccuAddressUnMgd.dll (unmanaged DLL) will call AccuAddress.dll (managed DLL) to perform an address lookup. You only need to do following to successfully integrate our AccuZIPToolkit for an unmanaged programming interface.

**Step 1:** Copy and Paste this logic into the global area of your DLL or EXE

```
typedef struct{
    TCHAR AddressLine1[50];
        TCHAR AddressLine2[50];
    TCHAR Company[50];
    TCHAR City[50];
    TCHAR State[20];
    TCHAR zipcode[10];
    TCHAR Urban[30];
}AccuInput;

typedef struct{
    char recNum[8+1];
    char iadl1[50+1];
    char iadl2[50+1];
    char icity[50+1];
    char iState[20+1];
    char iZipcode[10+1];
    char iPrUrb[28+1];
    char iadl3[50+1];
    char dadl3[50+1];
    char dadl1[50+1];
    char dadl2[50+1];
    char dLast[50+1];
    char dPrurb[28+1];
    char dCtys[28+1];
    char dStas[2+1];
    char dCtya[28+1];
    char dAbCty[13+1];
    char dStaa[2+1];
    char dZipc[5+1];
    char dAddon[4+1];
    char dDPBC[3+1];
    char dCris[4+1];
    char dCounty[30+1];
    char dResponse[2+1];
    char dRetcc[2+1];
    char dAdrkey[1+1];
    char dAutoZone[1+1];
    char dLotNum[4+1];
    char dLotCode[2+1];
    char dLacs_RC[2+1];
    char dLacs_indc[1+1];
    char pPrimaryNum[50+1];
    char pSecondaryNum[8+1];
    char pRoute[3+1];
    char pUnit[4+1];
    char pPre1[2+1];
    char pPre2[2+1];

    char pSuf1[4+1];
    char pSuf2[4+1];
    char pPst1[2+1];
    char pPst2[2+1];
    char pPrimaryName[28+1];
    char mPrimaryNum[10+1];
    char mSecondaryNum[8+1];
    char PMBUnit[3+1];
    char PMBNum[8+1];
    char FootNotes[32+1];
    char SuiteLink[2+1];
    char PUnit2[4+1];
    char PSNum2[8+1];
    char FootA[1+1];
    char FootB[1+1];
    char FootC[1+1];
    char FootD[1+1];
    char FootE[1+1];
    char FootF[1+1];
    char FootG[1+1];
    char FootH[1+1];
    char FootI[1+1];
    char FootJ[1+1];
    char FootK[1+1];
    char FootL[1+1];
    char FootM[1+1];
    char FootN[1+1];
    char FootO[1+1];
    char FootP[1+1];
    char FootQ[1+1];
    char FootR[1+1];
    char FootS[1+1];
    char FootT[1+1];
    char FootU[1+1];
    char FootV[1+1];
    char FootW[1+1];
    char FootX[1+1];
    char FootY[1+1];
    char FootZ[1+1];
    char Foot00[1+1];
    char Foot01[1+1];
    char Foot02[1+1];
    char Foot03[1+1];
    char Foot04[1+1];
    char Foot05[1+1];
    char Filler[1+1];
    char ZIP_CODE[5+1];
    char UPDATE_KEY[10+1];

    char Action[1+1];
    char Rec_Type[1+1];
    char Base[1+1];
    char Lacs[1+1];
    char Finance[6+1];
    char state[2+1];
    char County_Num[3+1];
    char Congress[2+1];
    char Muni[4+1];
    char Urban[4+1];
    char DPVHSA[1+1];
    char DPVHSC[1+1];
    char DPVHSF[1+1];
    char DPVHSV[1+1];
    char DPVHSX[1+1];
    char DPVAA[1+1];
    char DPVA1[1+1];
    char DPVBB[1+1];
    char DPVCC[1+1];
    char DPVN1[1+1];
    char DPVM1[1+1];
    char DPVM3[1+1];
    char DPVP1[1+1];
    char DPVP3[1+1];
    char DPVRR[1+1];
    char DPVR1[1+1];
    char DPVF1[1+1];
    char DPVG1[1+1];
    char DPVU1[1+1];
    char Misc[50+1];
}AccuOut;

HINSTANCE _createInstance;
typedef UINT (CALLBACK*
LPFNDFUNCLOOKUP)(AccuInput*,
AccuOut*);
LPFNDFUNCLOOKUP
lpfnDllFuncCASSLookup;
typedef UINT (CALLBACK*
LPFNDFUNCINIT)(BSTR);
LPFNDFUNCINIT lpfnDllFuncInit;

typedef UINT (CALLBACK*
LPFNDFUNC_CLOSE)();
LPFNDFUNC_CLOSE lpfnDllFuncClose;

HMODULE unmanagedLib;
```

**Step 2:** Copy this logic into Init function (i.e. OnInitDialog())

```
//Load Library\
unmangedLib = LoadLibraryA((LPCTSTR) "AccuAddressUnMgd.dll");

//This function will initialized AccuAddress COM dll
lpfnDllFuncInit =
(LPFNDDLLFUNCINIT)GetProcAddress(unmangedLib,LPCSTR("Init"));

//This function will lookup the address
lpfnDllFuncCASSLookup = (LPFNDDLLFUNCLOOKUP)GetProcAddress(unmangedLib,LPCSTR("AccuCassLookup"));

//This function will call AccuAddress COM DLL Close function
lpfnDllFuncClose=
(LPFNDDLLFUNC_CLOSE)GetProcAddress(unmangedLib,LPCSTR("Close"));

//Append "config.acu" file path.
BSTR configFile = SysAllocString(L"PathTo:\Config.acu");
lpfnDllFuncInit(configFile);
SysFreeString(configFile);
```

**Step 3:** This logic should appear in your lookup Event (i.e. OnBnClickedButton1)

Initialized input and output structure

```
AccuInput accln;
AccuOut accOut;
```

```
//Assign input data into your accln struture.
//This is just a sample code that reads input address from form
GetDlgItemText(IDC_EDIT1, accln.Company, sizeof(accln.Company) * sizeof(TCHAR));
GetDlgItemText(IDC_EDIT2, accln.Urban, sizeof(accln.Urban) * sizeof(TCHAR));
GetDlgItemText(IDC_EDIT3, accln.AddressLine1, sizeof(accln.AddressLine1) * sizeof(TCHAR));
GetDlgItemText(IDC_EDIT4, accln.AddressLine2, sizeof(accln.AddressLine2) * sizeof(TCHAR));
GetDlgItemText(IDC_EDIT5, accln.City, sizeof(accln.City) * sizeof(TCHAR));
GetDlgItemText(IDC_EDIT6, accln.State, sizeof(accln.State) * sizeof(TCHAR));
GetDlgItemText(IDC_EDIT7, accln.zipcode, sizeof(accln.zipcode) * sizeof(TCHAR));
```

CALL Address Lookup function

```
lpfnDllFuncCASSLookup(&accln,&accOut);
```

Now accuOut structure will have all the output fields updated.

**Step 4:** Call AccuAddress Close function

```
lpfnDllFuncClose();
```

Note: Only call this function once per session. Do not call this function after each lookup.





# AccuZIP Data Quality Toolkit Function Calls (Managed interface)

(Assuming AddrCode is the object reference to the COM AccuAddress.dll)

## Prerequisite

### Instantiate AccuZIPToolkit Object

```
AddrCode = CreateObject("AccuAddress.AddrCode6")
```

### Initialize and Open the AccuZIPToolkit Database

```
nResult=AddrCode.zp4connect("PathTo\Config.acu", ErrorMessage)
```

or

```
nResult=AddrCode.zp4connectalt("PathTo\Config.acu")
```

```
0      No errors, opened successfully
1      Error opening a file
2      Error reading a file
3      Error writing to Plus4cfg.dat file located in NATIONAL folder
4      Error finding a file
5      Registration Activation Number or Database has expired
6      Database files are out of sync
7      Security error
```

### Number of days until USPS Data Expires **(Optional call)**

```
nResult=AddrCode.zp4GetDataExpirationDays("PathTo\Config.acu", 0)
```

### Number of days until the AccuAddress.dll Expires **(Optional call)**

```
nResult=AddrCode.zp4GetDllExpirationDays("PathTo\Config.acu")
```

## Input Data

### Set Company Name

```
AddrCode.AZSetQuery_iadl2("Company Name")
```

### Set Address Line 1

```
AddrCode.AZSetQuery_iadl1("Primary Address")
```

### Set Address Line 2

```
AddrCode.AZSetQuery_iadl3("Secondary Address or Suite/Apt")
```

### Set city Name

```
AddrCode.AZSetQuery_ictyi("City")
```

### Set state

```
AddrCode.AZSetQuery_istai("State")
```

### Set ZIP Code **(Do not use for Canada)**

```
AddrCode.AZSetQuery_izipc("ZIP or ZIP+4")
```

### Set Urbanization

```
AddrCode.AZSetQuery_iprurb("Urbanization")
```

### Set Country **(Required Canada Only)**

```
AddrCode.AZSetQuery_icountry("CANADA")
```

### Set Canadian Postalcode **(Required Canada Only)**

```
AddrCode.AZSetQuery_ipostalcode("POSTAL CODE")
```

## Call Address Lookup

```
AddrCode.AZFindDeliverable()
```



## Result Data

### Original Input Firm/Company

AddrCode.AZGetResult\_iadl2(iadl2)

### Original Input Primary Address

AddrCode.AZGetResult\_iadl1(iadl1)

### Original Input Secondary Address

AddrCode.AZGetResult\_iadl3(iadl3)

### Original Input Input City

AddrCode.AZGetResult\_ictyi(ictyi)

### Original Input State Abbreviation

AddrCode.AZGetResult\_istai(istai)

### Original Input ZIP Code

AddrCode.AZGetResult\_izipc(izipc)

### Firm/Company

AddrCode.AZGetResult\_dadl2(dadl2)

### Standardized Primary Address

AddrCode.AZGetResult\_dadl1(dadl1)

### Secondary Address

AddrCode.AZGetResult\_dadl3(dadl3)

### Standardized Input City (if acceptable)

AddrCode.AZGetResult\_dctya(dctya)

### Standardized State Abbreviation

AddrCode.AZGetResult\_dstaa(dstaa)

### Standardized ZIP Code

AddrCode.AZGetResult\_zipc(zipc)

### Standardized Postalcode (Canada Only)

AddrCode.AZGetResult\_postalcode(postalcode)

### +4 Code

AddrCode.AZGetResult\_addon(addon)

### Delivery Point and Check Digit

AddrCode.AZGetResult\_dpbc(dpbc)

### Carrier Route Code

AddrCode.AZGetResult\_cris(cris)

### Line of Travel Number

AddrCode.AZGetResult\_elot\_num(lotnum)

### Line of Travel Asc/Desc Code

lotcode=AddrCode.AZGet\_elot\_code()

### Address Type

rec\_type=AddrCode.AZGetCharAddr\_rec\_type

### Base

AddrCode.AZGetResult\_base(base)

### Urbanization

AddrCode.AZGetResult\_dprurb(dprurb)

### State FIPS Code

AddrCode.AZGetAddr\_state(state)

### County FIPS Number

AddrCode.AZGetAddr\_county\_no(county\_no)

### County Name

AddrCode.AZGetAddr\_county\_name(county\_name)

### Congressional District Number

AddrCode.AZGetAddr\_congress\_dist(congress)

### Standardized Preferred City

AddrCode.AZGetResult\_dctys(dctys)

### Standardized Abbreviated City Name (if available)

AddrCode.AZGetResult\_abcty(abcty)

### Complete Standardized Last Line



AddrCode.AZGetResult\_dlast(dlast)

**LACS Match**

AddrCode.AZGetResult\_lacs(lacs)

**LACSLink Return Code**

AddrCode.AZGetLlkRc(lacs\_rc)

**LACSLink Indicator**

AddrCode.AZGetLlkInd(lacs\_ind)

**SuiteLink Code**

AddrCode.AZGetResult\_stelinkfoot(stelink)



## Delivery Point Confirmation Indicators

### DPV Confirmation Indicator

dpvhsa=AddrCode.AZGetDPVA()

Y =Address was DPV confirmed for both primary and (if present) secondary numbers.

D =Address was DPV confirmed for the primary number only, and Secondary number information was missing.

S = Address was DPV confirmed for the primary number only, and Secondary number information was present but unconfirmed.

N =Both Primary and (if present) Secondary number information failed to DPV Confirm.

### DPV CMRA Indicator

dpvhsc=AddrCode.AZGetDPVC

### DPV False Positive Indicator

dpvhst=AddrCode.AZGetDPVF

### DPV Vacancy Indicator

dpvhsv=AddrCode.AZGetDPVV

### DPV No Stats Indicator

dpvhst=AddrCode.AZGetDPVX

## Delivery Point Confirmation Footnotes

### AA Input Address Matched to the ZIP + 4 file

dpvaa=AddrCode.AZGet\_DPVfoot\_AA

### A1 Input Address Not Matched to the ZIP + 4 file

dpval=AddrCode.AZGet\_DPVfoot\_A1

### BB Input Address Matched to DPV (all components)

dpvbb=AddrCode.AZGet\_DPVfoot\_BB

### CC Input Address Primary Number Matched to DPV but Secondary Number not Matched (present but invalid)

dpvcc=AddrCode.AZGet\_DPVfoot\_CC

### N1 Input Address Primary Number Matched to DPV but Address Missing Secondary Number

dpvn1=AddrCode.AZGet\_DPVfoot\_N1

## Delivery Point Confirmation Footnotes (Continued)

### M1 Input Address Primary Number Missing

dpvm1=AddrCode.AZGet\_DPVfoot\_M1

### M3 Input Address Primary Number Invalid

dpvm3=AddrCode.AZGet\_DPVfoot\_M3

### P1 Input Address PO, RR, or HC Box number missing

dpvp1=AddrCode.AZGet\_DPVfoot\_P1

### P3 Input Address PO, RR, or HC Box number Invalid

dpvp3=AddrCode.AZGet\_DPVfoot\_P3

### RR Input Address Matched to CMRA and PMB designator present (PMB 123 or #123)

dpvrr=AddrCode.AZGet\_DPVfoot\_RR

### R1 Input Address Matched to CMRA but PMB designator not present (PMB 123 or #123)

dpvr1=AddrCode.AZGet\_DPVfoot\_R1

### F1 Input Address Matched to a Military Address

dpvf1=AddrCode.AZGet\_DPVfoot\_F1

### G1 Input Address Matched to a General Delivery Address

dpvg1=AddrCode.AZGet\_DPVfoot\_G1

### U1 Input Address Matched to a Unique ZIP Code

dpvul=AddrCode.AZGet\_DPVfoot\_U1



## **Parsed Address Information:**

### **Parsed Primary Number**

```
AddrCode.AZGetResult_mpnum(mpnum)
```

### **Parsed Primary Number if "mpnum" is empty**

```
AddrCode.AZGetResult_ppnum(ppnum)
```

### **Parsed Pre-direction**

```
AddrCode.AZGetAddr_pre_dir(pre_dir)
```

### **Parsed Pre-direction if "pre\_dir" is empty**

```
AddrCode.AZGetResult_pprel(pprel)
```

### **Parsed Street Name**

```
AddrCode.AZGetAddr_str_name(str_name)
```

### **Parsed Suffix**

```
AddrCode.AZGetAddr_suffix(suffix)
```

### **Parsed Suffix 1 if "suffix" is empty**

```
AddrCode.AZGetResult_psuf1(psuf1)
```

### **Parsed Suffix 2 if "suffix" is empty**

```
AddrCode.AZGetResult_psuf2(psuf2)
```

### **Parsed Post Direction**

```
AddrCode.AZGetAddr_post_dir(post_dir)
```

### **Parsed Post Direction if "post\_dir" is empty**

```
AddrCode.AZGetResult_ppst2(ppst2)
```

### **Parsed Unit Description**

```
AddrCode.AZGetAddr_unit(unit)
```

### **Parsed Unit Description if "unit" is empty**

```
AddrCode.AZGetResult_punit(punit)
```

### **Parsed Secondary Number**

```
AddrCode.AZGetResult_msnum(msnum)
```

### **Parsed Secondary Number if "msnum" is empty**

```
AddrCode.AZGetResult_psnum(psnum)
```

### **Private Mail Box Description**

```
AddrCode.AZGetResult_pmb(pmbunit)
```

### **Private Mail Box Number**

```
AddrCode.AZGetResult_pmbnum(pmbnum)
```

## **Set Abbreviate Address Flag:**

```
// Turns Standardized Street Name Abbreviation OFF
```

```
AddrCode.AZSetAbbrFlag(0)
```

```
// Turns Standardized Street Name Abbreviation ON
```

```
AddrCode.AZSetAbbrFlag(1)
```

Note: Call this only once after AddrCode.zp4connect()

## **Set Mixed Case ON and OFF:**

```
// Set Mixed Case ON
```

```
AddrCode6.AZSetCaseMixed(1)
```

```
// Set Mixed Case OFF, use Uppercase
```

```
AddrCode6.AZSetCaseMixed(0)
```

## **Closing AccuZIPToolkit Databases and Engine:**

```
AddrCode.zp4close()
```

## Warning and Response Values:

### Warnings or Errors

footnotes=AddrCode.AZGetFootnotes

A# ZIP  
B# City/State Corrected  
C# Invalid city/state/zip  
D# No ZIP assigned  
E# ZIP assigned for multiple response  
F# No ZIP available  
G# Part of firm moved to address  
H# Secondary number missing  
I# Insufficient/incorrect data  
J# Dual input  
K# Multi caused by cardinal rule  
L# Deliver address component add/del/chg  
M# Street name spelling changed  
N# Delivery address was standardized  
O# Low +4 tie-breaker (multi-response)  
P# Better delivery address exists  
Q# Unique ZIP Code  
R# No match due to EWS (Early Warning System)  
S# Invalid secondary number  
T# Multiple caused by magnet rule  
U# Unofficial Post Office name  
V# Unverifiable city/state  
W# Small town default  
X# Unique ZIP code generated  
Y# Military match  
Z# ZIP move match

### Number of responses

respn=AddrCode.AZGet\_respn()

### Return Code

retcc=AddrCode.AZGet\_retcc()

10 = Invalid Address  
11 = Invalid ZIP code  
12 = Invalid State Code  
13 = Invalid City  
21 = Address not found  
22 = Multiple response  
31 = Single response (Exact Match)  
32 = Default response (Missing information - Ste #, or Invalid Ste #)

### CASS Object Version #

retcc=AddrCode.zp4version()

### Residential Delivery Indicator (RDI)

AddrCode.AZGetAddr\_rdi(rdi\_flag)

Y = Residential Delivery  
N = Not Residential Delivery  
Blank = Did not query RDI



# “GET” Function Calls for environments that do not support “byRef”

If your programming environment does not support “byRef”, the function calls below can be used instead.

## Result Data

### Original Input Firm/Company

iadl2=AddrCode.AZGetResult\_iadl2\_AX()

### Original Input Primary Address

iadl1=AddrCode.AZGetResult\_iadl1\_AX()

### Original Input Secondary Address

iadl3=AddrCode.AZGetResult\_iadl3\_AX()

### Original Input Input City

ictyi=AddrCode.AZGetResult\_ictyi\_AX()

### Original Input State Abbreviation

istai=AddrCode.AZGetResult\_istai\_AX()

### Original Input ZIP Code

izipc=AddrCode.AZGetResult\_izipc\_AX()

### Firm/Company

dadl2=AddrCode.AZGetResult\_dadl2\_AX()

### Standardized Primary Address

dadl1=AddrCode.AZGetResult\_dadl1\_AX()

### Secondary Address

dadl3=AddrCode.AZGetResult\_dadl3\_AX()

### Standardized Input City (if acceptable)

dctya=AddrCode.AZGetResult\_dctya\_AX()

### Standardized State Abbreviation

dstaa=AddrCode.AZGetResult\_dstaa\_AX()

### Standardized ZIP Code

zipc=AddrCode.AZGetResult\_zipc\_AX()

### Standardized Postalcode (Canada Only)

zipc=AddrCode.AZGetResult\_postalcode\_AX()

### +4 Code

addon=AddrCode.AZGetResult\_addon\_AX()

### Delivery Point and Check Digit

dpbc=AddrCode.AZGetResult\_dpbc\_AX()

### Carrier Route Code

cris=AddrCode.AZGetResult\_cris\_AX()

### Line of Travel Number

lotnum=AddrCode.AZGetResult\_elot\_num\_AX()

### Line of Travel Asc/Desc Code

lotcode=AddrCode.AZGet\_elot\_code\_AX()

### Address Type

rec\_type=AddrCode.AZGetCharAddr\_rec\_type

### Base

base=AddrCode.AZGetResult\_base\_AX()

### Urbanization

dprurb=AddrCode.AZGetResult\_dprurb\_AX()

### State FIPS Code

state=AddrCode.AZGetAddr\_state\_AX()

### County FIPS Number

county\_no=AddrCode.AZGetAddr\_county\_no\_AX()

### County Name

county\_name=AddrCode.AZGetAddr\_county\_name\_AX()

### Congressional District Number

congress=AddrCode.AZGetAddr\_congress\_dist\_AX()

### Standardized Preferred City

dctys=AddrCode.AZGetResult\_dctys\_AX()



**Standardized Abbreviated City Name (if available)**

abcty=AddrCode.AZGetResult\_abcty\_AX()

**Complete Standardized Last Line**

dlast=AddrCode.AZGetResult\_dlast\_AX()

**LACS Match**

lacs=AddrCode.AZGetResult\_lacs\_AX()

**LACSLink Return Code**

lacs\_rc=AddrCode.AZGetLlkRc\_AX()

**LACSLink Indicator**

lacs\_ind=AddrCode.AZGetLlkInd\_AX()

**SuiteLink Code**

stelink=AddrCode.AZGetResult\_stelinkfoot\_AX()

**Delivery Point Confirmation Indicators****DPV Confirmation Indicator**

dpvhsa=AddrCode.AZGetDPVA()

Y =Address was DPV confirmed for both primary and (if present) secondary numbers.

D =Address was DPV confirmed for the primary number only, and Secondary number information was missing.

S = Address was DPV confirmed for the primary number only, and Secondary number information was present but unconfirmed.

N =Both Primary and (if present) Secondary number information failed to DPV Confirm.

**DPV CMRA Indicator**

dpvhsc=AddrCode.AZGetDPVC

**DPV False Positive Indicator**

dpvhst=AddrCode.AZGetDPVF

**DPV Vacancy Indicator**

dpvhsv=AddrCode.AZGetDPVV

**DPV No Stats Indicator**

dpvhstx=AddrCode.AZGetDPVX

**Delivery Point Confirmation Footnotes****AA Input Address Matched to the ZIP + 4 file**

dpvaa=AddrCode.AZGet\_DPVfoot\_AA

**A1 Input Address Not Matched to the ZIP + 4 file**

dpva1=AddrCode.AZGet\_DPVfoot\_A1

**BB Input Address Matched to DPV (all components)**

dpvbb=AddrCode.AZGet\_DPVfoot\_BB

**CC Input Address Primary Number Matched to DPV but Secondary Number not Matched (present but invalid)**

dpvcc=AddrCode.AZGet\_DPVfoot\_CC

**N1 Input Address Primary Number Matched to DPV but Address Missing Secondary Number**

dpvn1=AddrCode.AZGet\_DPVfoot\_N1

**Delivery Point Confirmation Footnotes (Continued)****M1 Input Address Primary Number Missing**

dpvm1=AddrCode.AZGet\_DPVfoot\_M1

**M3 Input Address Primary Number Invalid**

dpvm3=AddrCode.AZGet\_DPVfoot\_M3

**P1 Input Address PO, RR, or HC Box number missing**

dpvp1=AddrCode.AZGet\_DPVfoot\_P1

**P3 Input Address PO, RR, or HC Box number Invalid**

dpvp3=AddrCode.AZGet\_DPVfoot\_P3

**RR Input Address Matched to CMRA and PMB designator present (PMB 123 or #123)**

dpvrr=AddrCode.AZGet\_DPVfoot\_RR

**R1 Input Address Matched to CMRA but PMB designator not present (PMB 123 or #123)**

dpvr1=AddrCode.AZGet\_DPVfoot\_R1





### **F1 Input Address Matched to a Military Address**

```
dpvf1=AddrCode.AZGet_DPVfoot_F1
```

### **G1 Input Address Matched to a General Delivery Address**

```
dpvg1=AddrCode.AZGet_DPVfoot_G1
```

### **U1 Input Address Matched to a Unique ZIP Code**

```
dpvu1=AddrCode.AZGet_DPVfoot_U1
```

### **Parsed Address Information:**

#### **Parsed Primary Number**

```
mpnum=AddrCode.AZGetResult_mpnum_AX()
```

#### **Parsed Primary Number if "mpnum" is empty**

```
ppnum=AddrCode.AZGetResult_ppnum_AX()
```

#### **Parsed Pre-direction**

```
pre_dir=AddrCode.AZGetAddr_pre_dir_AX()
```

#### **Parsed Pre-direction if "pre\_dir" is empty**

```
pprel=AddrCode.AZGetResult_pprel_AX()
```

#### **Parsed Street Name**

```
str_name=AddrCode.AZGetAddr_str_name_AX()
```

#### **Parsed Suffix**

```
suffix=AddrCode.AZGetAddr_suffix_AX()
```

#### **Parsed Suffix 1 if "suffix" is empty**

```
psuf1=AddrCode.AZGetResult_psuf1_AX()
```

#### **Parsed Suffix 2 if "suffix" is empty**

```
psuf2=AddrCode.AZGetResult_psuf2_AX()
```

#### **Parsed Post Direction**

```
post_dir=AddrCode.AZGetAddr_post_dir_AX()
```

#### **Parsed Post Direction if "post\_dir" is empty**

```
ppst2=AddrCode.AZGetResult_ppst2_AX()
```

#### **Parsed Unit Description**

```
unit=AddrCode.AZGetAddr_unit_AX()
```

#### **Parsed Unit Description if "unit" is empty**

```
Punit=AddrCode.AZGetResult_punit_AX()
```

#### **Parsed Secondary Number**

```
msnum=AddrCode.AZGetResult_msnum_AX()
```

#### **Parsed Secondary Number if "msnum" is empty**

```
psnum=AddrCode.AZGetResult_psnum_AX()
```

#### **Private Mail Box Description**

```
pmbunit=AddrCode.AZGetResult_pmb_AX()
```

#### **Private Mail Box Number**

```
pmbnum=AddrCode.AZGetResult_pmbnum_AX()
```

### **Set Abbreviate Address Flag:**

```
// Turns Standardized Street Name Abbreviation OFF
```

```
AddrCode.AZSetAbbrFlag(0)
```

```
// Turns Standardized Street Name Abbreviation ON
```

```
AddrCode.AZSetAbbrFlag(1)
```

Note: Call this only once after AddrCode.zp4connect()

### **Closing AccuZIPToolkit Databases and Engine:**

```
AddrCode.zp4close()
```



## Warning and Response Values:

### Warnings or Errors

footnotes=AddrCode.AZGetFootnotes

A# ZIP  
B# City/State Corrected  
C# Invalid city/state/zip  
D# No ZIP assigned  
E# ZIP assigned for multiple response  
F# No ZIP available  
G# Part of firm moved to address  
H# Secondary number missing  
I# Insufficient/incorrect data  
J# Dual input  
K# Multi caused by cardinal rule  
L# Deliver address component add/del/chg  
M# Street name spelling changed  
N# Delivery address was standardized  
O# Low +4 tie-breaker (multi-response)  
P# Better delivery address exists  
Q# Unique ZIP Code  
R# No match due to EWS (Early Warning System)  
S# Invalid secondary number  
T# Multiple caused by magnet rule  
U# Unofficial Post Office name  
V# Unverifiable city/state  
W# Small town default  
X# Unique ZIP code generated  
Y# Military match  
Z# ZIP move match

### Number of responses

respn=AddrCode.AZGet\_respn()

### Return Code

retcc=AddrCode.AZGet\_retcc()

10 = Invalid Address  
11 = Invalid ZIP code  
12 = Invalid State Code  
13 = Invalid City  
21 = Address not found  
22 = Multiple response  
31 = Single response (Exact Match)  
32 = Default response (Missing information - Ste #, or Invalid Ste #)

### CASS Object Version #

retcc=AddrCode.zp4version()

### Residential Delivery Indicator (RDI)

rdi\_flag=AddrCode.AZGetAddr\_rdi\_AX()

# AccuZIP Data Quality Toolkit JSON Input/Output

(VB Example: Assuming AddrCode is the object reference to the COM AccuAddress.dll)

## Prerequisite

### Instantiate AccuZIPToolkit Object

```
AddrCode = CreateObject("AccuAddress.AddrCode6")
```

### Initialize and Open the AccuZIPToolkit Database

```
nResult=AddrCode.zp4connect("PathTo\Config.acu", ErrorMessage)
or
nResult=AddrCode.zp4connectalt("PathTo\Config.acu")
```

```
0    No errors, opened successfully
1    Error opening a file
2    Error reading a file
3    Error writing to Plus4cfg.dat file located in NATIONAL folder
4    Error finding a file
5    Registration Activation Number or Database has expired
6    Database files are out of sync
7    Security error
```

### Number of days until USPS Data Expires (Optional call)

```
nResult=AddrCode.zp4GetDataExpirationDays("PathTo\Config.acu", 0)
```

### Number of days until the AccuAddress.dll Expires (Optional call)

```
nResult=AddrCode.zp4GetDllExpirationDays("PathTo\Config.acu")
```

## Input Data (Single Address Sample)

```
/****** JSON INPUT DATA (jsonAddr) *****/
[
  { Address:
    { "AZSetQuery_irecid":"1",
      "AZSetQuery_iadl1":"943 Greenwood Ct",
      "AZSetQuery_iadl2":"",
      "AZSetQuery_iadl3":"",
      "AZSetQuery_icty":"trenton",
      "AZSetQuery_istai":"OH",
      "AZSetQuery_izipc":"45014",
      "AZSetQuery_iprurb":""
    }
  }
]
```

Note: You can pass multiple addresses by separating each address block with a comma, e.g.,  
[{ Address: {inputAddressBlock} },{Address: {inputAddressBlock} }] You can also use  
"AZSetQuery\_iforeignid" in place of "AZSetQuery\_irecid" if you need longer unique record IDs.

## Call Address Lookup

```
result = AddrCode6.AZProcessJSONAddr(jsonAddr)
```



## Result Data

/\*\*\*\* JSON OUTPUT DATA (result)\*\*\*\*/

```
[
  {"Addr Result":
    {
      "dadl3": "",
      "dadl1": "943 GREENWOOD CT",
      "dadl2": "",
      "dlast": "TRENTON OH 45067",
      "dctys": "TRENTON",
      "dstaa": "OH",
      "dctya": "",
      "abcty": "",
      "zipc": "45067",
      "addon": "1079",
      "dpbc": "434",
      "cris": "R001",
      "county_no": "017",
      "county_name": "BUTLER",
      "respn": "1",
      "retcc": "31",
      "elot_num": "0058",
      "elot_code": "D",
      "LlkRc": "",
      "LlkInd": "",
      "ppnum": "943",
      "psnum": "",
      "prote": "R001",
      "punit": "",
      "ppre1": "",
      "ppre2": "",
      "psuf1": "CT",
      "psuf2": "",
      "ppst1": "",
      "ppst2": "",
      "ppnam": "GREENWOOD",
      "mpnum": "943",
      "msnum": "",
      "pmb": "",
      "pmbnum": "",
      "Footnotes": "A#N#V#",
      "stelnkInd": "",
      "punit2": "",
      "psnum2": "",
      "A": "A",
      "B": "",
      "C": "",
      "D": "",
      "E": "",
      "F": "",
      "G": "",
      "H": "",
      "I": "",
      "J": "",
      "K": "",
      "L": "",
      "M": "",
      "N": "N",
      "O": "",
      "P": "",
      "Q": "",
      "R": "",
      "S": "",
      "T": "",
      "U": "",
      "V": "V",
      "W": "",
      "X": "",
      "Y": "",
      "Z": "",
      "00": "",
      "01": "",
      "02": "",
      "03": "",
      "04": "",
      "05": "",
      "rec_type": "S",
      "pre_dir": "",
      "str_name": "GREENWOOD",
      "suffix": "CT",
      "post_dir": "",
      "sec_name": "",
      "unit": "",
      "sec_low": "",
      "sec_high": "",
      "sec_code": "",
      "addon_low": "1079",
      "addon_high": "1079",
      "lacs": "",
      "finance": "388295",
      "state": "39",
      "congress_dist": "08",
      "municipality": "",
      "DPVA": "Y",
      "DPVC": "N",
      "DPVF": "",
      "DPVV": "N",
      "DPVX": "N",
      "DPVfoot_AA": "1",
      "DPVfoot_A1": "0",
      "DPVfoot_BB": "1",
      "DPVfoot_CC": "0",
      "DPVfoot_N1": "0",
      "DPVfoot_M1": "0",
      "DPVfoot_M3": "0",
      "DPVfoot_P1": "0",
      "DPVfoot_P3": "0",
      "DPVfoot_RR": "0",
      "DPVfoot_R1": "0",
      "DPVfoot_F1": "0",
      "DPVfoot_G1": "0",
      "DPVfoot_U1": "0",
      "misc": "",
      "irecid": "1",
      "iadl1": "943 Greenwood Ct",
      "iadl2": "",
      "icty": "trenton",
      "istai": "OH",
      "izipc": "45014",
      "iprurb": "",
      "iadl3": "",
      "rdi": "Y"
    }
  ]
```



## Input Data (Multiple Address Sample)

```
[
  {
    "Address": {
      "AZSetQuery_irecid": "1",
      "AZSetQuery_iadl1": "943 Greenwood Ct",
      "AZSetQuery_iadl2": "",
      "AZSetQuery_iadl3": "",
      "AZSetQuery_ictyi": "trenton",
      "AZSetQuery_istai": "OH",
      "AZSetQuery_izipc": "45014",
      "AZSetQuery_iprurb": ""
    }
  },
  {
    "Address": {
      "AZSetQuery_irecid": "1",
      "AZSetQuery_iadl1": "1600 Amphitheatre Parkway",
      "AZSetQuery_iadl2": "",
      "AZSetQuery_iadl3": "",
      "AZSetQuery_ictyi": "Mountain View",
      "AZSetQuery_istai": "CA",
      "AZSetQuery_izipc": "94043",
      "AZSetQuery_iprurb": ""
    }
  }
]
```



## Result Data: (Multiple Address Sample)

/\*\*\*\* JSON OUTPUT DATA (result)\*\*\*\*/

```
{
  "Addr_Result": [
    {
      "dadl3": "",
      "dadl1": "943 GREENWOOD CT",
      "dadl2": "",
      "dlast": "TRENTON OH 45067",
      "dctys": "TRENTON",
      "dstaa": "OH",
      "dctya": "",
      "abcty": "",
      "zipc": "45067",
      "addon": "1079",
      "dpbc": "434",
      "cris": "R001",
      "county_no": "017",
      "county_name": "BUTLER",
      "respn": "1",
      "retcc": "31",
      "elot_num": "0058",
      "elot_code": "D",
      "LlkRc": "",
      "LlkInd": "",
      "ppnum": "943",
      "psnum": "",
      "prote": "R001",
      "punit": "",
      "ppre1": "",
      "ppre2": "",
      "psuf1": "CT",
      "psuf2": "",
      "ppst1": "",
      "ppst2": "",
      "ppnam": "GREENWOOD",
      "mpnum": "943",
      "msnum": "",
      "pmb": "",
      "pmbnum": "",
      "Footnotes": "A#N#V#",
      "stelkInd": "",
      "punit2": "",
      "psnum2": "",
      "A": "A",
      "B": "",
      "C": "",
      "D": "",
      "E": "",
      "F": "",
      "G": "",
      "H": "",
      "I": "",
      "J": "",
      "K": "",
      "L": "",
      "M": "",
      "N": "N",
      "O": "",
      "P": "",
      "Q": "",
      "R": "",
      "S": "",
      "T": "",
      "U": "",
      "V": "",
      "W": "",
      "X": "",
      "Y": "",
      "Z": "",
      "00": "",
      "01": "",
      "02": "",
      "03": "",
      "04": "",
      "05": "",
      "rec_type": "S",
      "pre_dir": "",
      "str_name": "GREENWOOD",
      "suffix": "CT",
      "post_dir": "",
      "sec_name": "",
      "unit": "",
      "sec_low": "",
      "sec_high": "",
      "sec_code": "",
      "addon_low": "1079",
      "addon_high": "1079",
      "lacs": "",
      "finance": "388295",
      "state": "9",
      "congress_dist": "8",
      "municipality": "",
      "DPVA": "Y",
      "DPVC": "N",
      "DPVF": "",
      "DPVV": "N",
      "DPVX": "N",
      "DPVfoot_AA": "1",
      "DPVfoot_A1": "0",
      "DPVfoot_BB": "1",
      "DPVfoot_CC": "0",
      "DPVfoot_N1": "0",
      "DPVfoot_M1": "0",
      "DPVfoot_M3": "0",
      "DPVfoot_P1": "0",
      "DPVfoot_P3": "0",
      "DPVfoot_RR": "0",
      "DPVfoot_R1": "0",
      "DPVfoot_F1": "0",
      "DPVfoot_G1": "0",
      "DPVfoot_U1": "0",
      "misc": "",
      "irecid": "1",
      "iadl1": "943 Greenwood Ct",
      "iadl2": "",
      "ictyi": "trenton",
      "istai": "OH",
      "izipc": "45014",
      "ipurb": "",
      "iadl3": "",
      "rdi": "Y"
    },
    {
      "dadl3": "",
      "dadl1": "1600 AMPHITHEATRE
      PKWY",
      "dadl2": "",
      "dlast": "MOUNTAIN VIEW CA 94043",
      "dctys": "MOUNTAIN VIEW",
      "dstaa": "CA",
      "dctya": "",
      "abcty": "",
      "zipc": "94043",
      "addon": "1351",
      "dpbc": "000",
      "cris": "C058",
      "county_no": "085",
      "county_name": "SANTA CLARA",
      "respn": "1",
      "retcc": "31",
      "elot_num": "0004",
      "elot_code": "A",
      "LlkRc": "",
      "LlkInd": "",
      "ppnum": "1600",
      "psnum": "",
      "prote": "C058",
      "punit": "",
      "ppre1": "",
      "ppre2": "",
      "psuf1": "PKWY",
      "psuf2": "",
      "ppst1": "",
      "ppst2": "",
      "ppnam": "AMPHITHEATRE",
      "mpnum": "1600",
      "msnum": "",
      "pmb": "",
      "pmbnum": "",
      "Footnotes": "N#",
      "stelkInd": "",
      "punit2": "",
      "psnum2": "",
      "A": "",
      "B": "",
      "C": "",
      "D": "",
      "E": "",
      "F": "",
      "G": "",
      "H": "",
      "I": "",
      "J": "",
      "K": "",
      "L": "",
      "M": "",
      "N": "N",
      "O": "",
      "P": "",
      "Q": "",
      "R": "",
      "S": "",
      "T": "",
      "U": "",
      "V": "",
      "W": "",
      "X": "",
      "Y": "",
      "Z": "",
      "00": "",
      "01": "",
      "02": "",
      "03": "",
      "04": ""
    }
  ]
}
```



```
"05": "",
"rec_type": "S",
"pre_dir": "",
"str_name": "AMPHITHEATRE",
"suffix": "PKWY",
"post_dir": "",
"sec_name": "",
"unit": "",
"sec_low": "",
"sec_high": "",
"sec_code": "",
"addon_low": "1351",
"addon_high": "1351",
"lacs": "",
"finance": "055208",
"state": "6",
"congress_dist": "8",
```

```
"municipality": "",
"DPVA": "Y",
"DPVC": "N",
"DPVF": "",
"DPVV": "N",
"DPVX": "N",
"DPVfoot_AA": "1",
"DPVfoot_A1": "0",
"DPVfoot_BB": "1",
"DPVfoot_CC": "0",
"DPVfoot_N1": "0",
"DPVfoot_M1": "0",
"DPVfoot_M3": "0",
"DPVfoot_P1": "0",
"DPVfoot_P3": "0",
"DPVfoot_RR": "0",
"DPVfoot_R1": "0",
```

```
"DPVfoot_F1": "0",
"DPVfoot_G1": "0",
"DPVfoot_U1": "0",
"misc": "",
"irecid": "1",
"iadl1": "1600 Amphitheatre
Parkway",
"iadl2": "",
"ictyi": "Mountain View",
"istai": "CA",
"izipc": "94043",
"ipurb": "",
"iadl3": "",
"rdi": "N"
}
]
```

