AccuZIPToolkit

Data Quality Developer Toolkit

AccuZIP Data Quality Developer Toolkit
Attention: This product incorporates a "disabling clock" feature designed to deactivate all functionalities of the AccuZIP Data Quality Toolkit in the event that the database no longer complies with the current Postal Service requirements. To ensure uninterrupted utilization of the toolkit, it is imperative that updates for the upcoming period are promptly acquired.

Calling the AccuZIP Data Quality Toolkit from Other Applications
The AccuZIP Data Quality Toolkit offers a comprehensive interface that facilitates the seamless integration of the toolkit's functionalities within programmers' applications.

Through the utilization of the Developer's Toolkit, applications gain the capability to invoke the Toolkit's functions specifically designed for Address Standardization. However, it is essential to ensure that the input data (including Company, Address, Address2, City, State, ZIP Code, and Urbanization) adheres to the prescribed format outlined below. Likewise, the output format for Standardized addresses is also detailed below.

The AccuZIP Data Quality Developer Toolkit is compatible with the dBase programming language, as well as other popular languages such as Visual Basic, C#, C++, C, FoxPro, and Java. Notably, the Toolkit operates transparently when utilized within any running application.

Getting Started

Step 1

To install the AccuZIP Data Quality Toolkit, please execute the Setup.exe file from the AccuZIP Data Quality Toolkit DVD or downloaded package. This installation procedure will incorporate the following files into your system:

Program Files

AccuAddress.dll – CASS Certified .NET component used to perform address matching.

AccuAddress.tlb – When associated with a DLL (Dynamic Link Library), a TLB (Type Library) is a binary file that contains information about the exposed interfaces, classes, and data structures within the DLL. It serves as a crucial component in facilitating communication and interoperability between different programming languages and environments.

A TLB file provides a standardized interface definition, allowing developers to understand the structure and capabilities of the DLL without needing access to the source code. It defines the methods, properties, and data types that can be accessed and utilized by external applications or components that interact with the DLL.

By referencing the TLB file, developers can easily integrate the DLL into their projects, ensuring proper interaction and utilization of the exposed functionality. This promotes interoperability and simplifies the process of working with DLLs across different programming languages and environments.

AddrCode.dbf – The output structure, i.e., AddrCode.dbf for AccuAddress.dll must be consistently updated with each release to ensure optimal performance. It is imperative to always maintain the latest version of this file.

Plus4cfg.dat – Database Integrity and Security File

Config.acu – Path information to the database files

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

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

Database files:

ADRDATA folder

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

Step 2

The "Activate AccuZIPToolkit" dialog box will appear when the zp4connect(path, msg) function is called for the first time or if the product has expired. In this dialog box, please enter the Registration Serial Number that was provided to you in the initial email for the AccuZIPToolkit. You can either type it manually or copy/paste it.

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 Adddress
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 Adddress
CASS. will have list of 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 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  PUnite2[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  DPVR7[1+1];
   char  DPVPB[1+1];
   char  Misc[50+1];
   char  Misc[30+1];
   char  ForeignID[36+1];
   char  iCountry[30+1];
   char  PostalCode[16+1];
   char  DPVEXT[1+1];
   char  DPVTA[1+1];
   char  DPVDN[1+1];
   char  DPVNS[1+1];
   char  DPVND[1+1];
   char  DPVX1[1+1];
   char  DPVX2[1+1];
   char  DPVX3[1+1];
   char  DPVX4[1+1];
   char  DPVX5[1+1];
   char  DPVX6[1+1];
   char  DPVX7[1+1];
   char  DPV01[1+1];
   char  DPV02[1+1];
   char  DPV03 [1+1];
   char  DPV04 [1+1];
   char  DPV05 [1+1];
   char  DPV06 [1+1];
   char  DPVPO [1+1];
   char  DPVC1[1+1];
   char  DPVIA[1+1];
   char  FOOTIA[1+1];
   char  DPVHSD[1+1];
}AccuOut;

HINSTANCE _createInstance;
typedef UINT (CALLBACK*
LPFNDLLFUNCLOOKUP)(AccuInput*, AccuOut*);
LPFNDLLFUNCLOOKUP lpfnDllFuncCASSLookup;
typedef UINT (CALLBACK*
LPFNDLLFUNCINIT)(BSTR);
LPFNDLLFUNCINIT lpfnDllFuncInit;

typedef UINT (CALLBACK*
LPFNDLLFUNCCLOSE)();
LPFNDLLFUNCCLOSE lpfnDllFuncClose;
```

```
HMODULE  unmangedLib;
```

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

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

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

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

//This function will call AccuAddress COM DLL Close function
lpfnDllFuncClose=
(LPFNDLLFUNCCLOSE)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 accIn;
AccuOut accOut;

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

CALL Address Lookup function
lpfnDllFuncCASSLookup(&accIn,&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", ErrorMsg)
```

0	No errors, opened successfully
1	Error opening a file
2	Error reading a file
3	Error writing to a file
4	Error finding a file
5	Library has expired
6	Database files are out of sync
7	Security error

Number of days until USPS Data Expires (Optional call)

```
nResult=AddrCode.zp4GetDataExpirationDay("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

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

Set Urbanization

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

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)
+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
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 Abreviated 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_stelnkfoot(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

dpvhsf=AddrCode.AZGetDPVF

DPV Vacancy Indicator

dpvhsv=AddrCode.AZGetDPVV

DPV No Stats Indicator

dpvhsx=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, secondary number not matched; secondary number not required

dpvcc=AddrCode.AZGet_DPVfoot_CC

C1 Input address primary number matched, secondary number not matched; secondary number required

Dpvc1=AddrCode.AZGet_DPVfoot_C1

N1 Input address primary number matched to DPV® but address missing required 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

R7 Addresses that are assigned to a phantom route of R777 or R779

Dpvr7=AddrCode.AZGet_DPVfoot_R7

PB Identified PO Box Street Address

Dpvpb=AddrCode.AZGet_DPVfoot_PB

DNA Door Not Accessible

Dpvdn=AddrCode.AZGet_DPVfoot_DN

NSL No Secure Location

Dpvns=AddrCode.AZGet_DPVfoot_NS

ND Non Delivery Days

Dpvnd=AddrCode.AZGet_DPVfoot_ND

PO PO Box™ Throwback Table

Dpvpo=AddrCode.AZGet_DPVfoot_PO

IA Informed Address identified

Dpvia=AddrCode.AZGet_DPVfoot_IA
TA Input Address Primary Number Matched to DPV® by dropping trailing alpha
Dpvia=AddrCode.AZGet_DPVfoot_IA
X1 Non Delivery Days - Monday
Dpvx1=AddrCode.AZGet_DPVfoot_X1
X2 Non Delivery Days - Tuesday
Dpvx2=AddrCode.AZGet_DPVfoot_X2
X3 Non Delivery Days - Wednesday
Dpvx3=AddrCode.AZGet_DPVfoot_X3
X4 Non Delivery Days - Thursday
Dpvx4=AddrCode.AZGet_DPVfoot_X4
X5 Non Delivery Days - Friday
Dpvx5=AddrCode.AZGet_DPVfoot_X5
X6 Non Delivery Days - Saturday
Dpvx6=AddrCode.AZGet_DPVfoot_X6
X7 Non Delivery Days - Sunday
Dpvx7=AddrCode.AZGet_DPVfoot_X7
01 NoStat Reason Code - Internal Drop Address. These are addresses that do not receive mail delivery directly from the USPS, but are delivered to a drop address that services them.
Dpv01=AddrCode.AZGet_DPVfoot_01
02 NoStat Reason Code - The delivery is new construction and delivery has not been established or on a Rural/CDS/HCR where the delivery point is unoccupied for more than 90 days.
Dpv02=AddrCode.AZGet_DPVfoot_02
03 NoStat Reason Code – Collision – These addresses do not actually DPV® confirm.
Dpv03=AddrCode.AZGet_DPVfoot_03
04 NoStat Reason Code - College/Military Zone, & Other types. These are ZIP + 4 ® records USPS has incorporated into the data.
Dpv04=AddrCode.AZGet_DPVfoot_04
05 NoStat Reason Code - The address is no longer a possible delivery, the address is on an R777 route, or the PO Box has never been rented or has been declared un rentable, or the No Stat reason is not one of the other reasons.
Dpv05=AddrCode.AZGet_DPVfoot_05
06 NoStat Reason Code - The address requires secondary information.
Dpv06=AddrCode.AZGet_DPVfoot_06

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_ppre1(ppre1)
Parsed Street Name
AddrCode.AZGetAddr_str_name(str_name)
Parsed Suffix
AddrCode.AZGetAddr_suffix(suffix)
Parsed Suffix 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()

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 = Invald 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

Note: You must contact the USPS® to obtain a license to the RDI data by completing the RDI Product Application (https://ribbs.usps.gov/rdi/documents/tech_guides/RDI_APPLICATION.PDF)

Once you receive the RDI data from the USPS, place the R9 and R11 files inside the \ADRDATA\LACSLink\ folder. Then when the AddrCode.AZGetAddr_rdi() function is called, the RDI Flag will be returned.