



## .NET TOOLKIT

### WHAT IS IT?




“.NET is the *Microsoft* Web services strategy to connect information, people, systems, and devices through software. .NET-connected solutions enable businesses to integrate their systems more rapidly and in a more agile manner and help them realize the promise of information anytime, anywhere, on any device.”- Microsoft<sup>®</sup>

The **AccuMail .NET Toolkit** is a developer's toolkit designed to create a connection from AccuMail frameworks to your own software applications, allowing address validation and correction capabilities to be easily integrated into your organization's existing software applications, or even into your company website or intranet. It is a simple **Application Program Interface (API)** that can be called from your application from a variety of languages, including:

-  *PHP (Web Applications)*
-  *C# (C Sharp)*
-  *VB.NET (Visual Basic .NET)*

### EXAMPLES OF USE




Incorporate the power of AccuMail address correction into your:

-  **Websites** – validate customer address data as it is entered eg for eCommerce transactions or account creation.
-  **Shipping and Fulfillment Systems** – verifying addresses prior to shipping helps eliminate expensive repackaging/reshipping costs, as well as address correction penalties charged by carriers.
-  **CRM Applications** – validating addresses as they are entered into your CRM database prevents inaccurate or incomplete data causing problems throughout your organization.



### WHAT'S IN THE TOOLKIT?

The .NET Toolkit contains the following components:

-  *.NET Toolkit documentation*
-  *C# sample code*
-  *VB.net sample code*



# AccuMail Toolkits

Integrate The Power of AccuMail



AccuMail  
Certified™

## IMPLEMENTATION

The .NET Toolkit is both easy to use and simple to install, with a typical installation taking as little as an hour or two. Below is a sample:

### Step One: Reference AccuMailGoldDotNet.dll

In Visual Studios, add a reference to the AccuMailGoldDotnet.dll

### Step Two: Instance of AccuMailGoldDotNet

```
private AccuMailGoldDotNet.Z4Object objAm;
private IContainer components;
```

### Step Three: (Form Constructor) A call to SetControlStates must be added to the constructor to set the initial state of the controls on the form (Req for Windows Form Designer support)

```
InitializeComponent();
SetControlStates();
```

### Step Four: call the Z4Object.Initialize method which opens the AccuMail database

```
objAm = new AccuMailGoldDotNet.Z4Object();
String accumailPath = txtAccuMailPath.Text;
objAm.Initialize(txtAccuMailPath.Text, "National");
objAm.Initialize();
```

### Step Five: Perform Lookup

```
LookupAddress()
objAm.ClearBuffers();
SetAccuMailSettings();
```

```
objAm.SetField(Z4Field.Z4_COMPANY, txtCompany.Text);
objAm.SetField(Z4Field.Z4_STREET, txtStreet.Text);
objAm.SetField(Z4Field.Z4_SUITE, txtSuite.Text);
objAm.SetField(Z4Field.Z4_STREET2, txtStreet2.Text);
objAm.SetField(Z4Field.Z4_URB, txtUrb.Text);
objAm.SetField(Z4Field.Z4_CITY, txtCity.Text);
objAm.SetField(Z4Field.Z4_STATE, txtState.Text);
objAm.SetField(Z4Field.Z4_ZIP, txtZip.Text);
objAm.SetField(Z4Field.Z4_ZIP4, txtZip4.Text);
```

```
if (objAm.Lookup())
txtCompany.Text = objAm.GetField(Z4Field.Z4_COMPANY);
txtStreet.Text = objAm.GetField(Z4Field.Z4_STREET);
txtSuite.Text = objAm.GetField(Z4Field.Z4_SUITE);
txtStreet2.Text = objAm.GetField(Z4Field.Z4_STREET2);
txtCity.Text = objAm.GetField(Z4Field.Z4_CITY);
txtState.Text = objAm.GetField(Z4Field.Z4_STATE);
txtZip.Text = objAm.GetField(Z4Field.Z4_ZIP);
txtZip4.Text = objAm.GetField(Z4Field.Z4_ZIP4);
```

### Step Six: Use GetAddress and SetAddress methods to lookup an Address

```
lookupAddressUsingGetSetAddress()
objAm.ClearBuffers();
SetAccuMailSettings();
AccuMailGoldDotNet.InputAddress ia = new InputAddress();
```

```
ia.Company = txtCompany.Text;
ia.Street = txtStreet.Text;
ia.Suite = txtSuite.Text;
ia.Street2 = txtStreet2.Text;
ia.Urbanization = txtUrb.Text;
ia.City = txtCity.Text;
ia.State = txtState.Text;
ia.Zip = txtZip.Text;
ia.Plus4 = txtZip4.Text;
```

```
objAm.SetAddress(ia);
```

```
if (objAm.Lookup())
AccuMailGoldDotNet.OutputAddress oa = objAm.GetAddress();
txtCompany.Text = oa.Company;
txtStreet.Text = oa.Street;
txtSuite.Text = oa.Suite;
txtStreet2.Text = oa.Street2;
txtCity.Text = oa.City;
txtState.Text = oa.State;
txtZip.Text = oa.Zip;
txtZip4.Text = oa.Plus4;
```

### Step Seven: Method to close an open connection to AccuMail

```
objAm.CloseAccuMail();
```

## SYSTEM REQUIREMENTS

### Recommended:

Pentium 3 processor or higher  
512 MB RAM minimum, 2GB optimal  
400 MB Free Disk Space  
Windows XP or higher OS  
Microsoft IIS 4.0 or higher for ASP page samples  
.NET Frameworks 2.0 or higher

For more information on the .NET Toolkit or to receive a free sample of code for evaluation purposes, please contact us at [sales@smartsoftusa.com](mailto:sales@smartsoftusa.com) or call us on 888.227.7221.