

Integrating Visual FoxPro and MailChimp – Part 2

Whil Hentzen

We've all written our own email applications. I finally decided to use an outside service to handle my emailing needs. Here's how I used VFP to integrate with the mailing service.

The first article in this series described how to set up MC to manage your mailing list, and then how to coordinate your current VFP mailing list with your new mailing list in MC.

Admittedly, it was a bit simplistic, I don't imagine too many of you were scratchign your head, syaing, Whoa, slow down. But it provided needed background.

As I mentioend last article, MC has an API that allows you to interact automatically with your data and drive the various functionality available on their site. The MC API exposes a bunch of functions, grouped in four categories: Export, List Management, Campaign Creation, and Campaign Statistics.

For example, you can download your list, unsubscribe users and get stats about your lists. All from VFP, rather than having to manually go to their website.

In this article, I'll show you how to perform each of those functions from VFP.

Connecting to the Web with VFP

Since we're going to be communicating with the MC website, VFP by itself isn't enough. We'll need a set of tools for Internet connections; in this article, I'll use West-Wind's Client Tools, available here.

<http://store.west-wind.com/product/WWCLIENT50>

Unzip the wwclient.zip file into a folder (I keep mine in a 'wwclient' folder under my 'dev' folder. Then load up VFP and include

```
F:\DROPOBOX\DROPOBOX\DEV\WWCLIENT;  
\DROPOBOX\DEV\WWCLIENT\CLASSES;  
\DROPOBOX\DEV\WWCLIENT\TOOLS
```

in your path. Then

```
do classes\wwclient  
do wwhttp  
o=createobject('wwhttp')
```

and you're ready to go.

How the MC API works

The MC API consists of a set of URLs. You choose the URL applicable for a specific function (say, downloading a list) and include parameters that identify your account and list. Executing the URL will perform an action on the MC server and, in some cases, return a value that represents the data you're looking for.

The MC API URLs have the following general format:

```
https://<dc>.api.mailchimp.com/2.0/<specific  
api call>?<parameters>
```

Let's look at the three variable parts.

<dc>

MC runs several data centers, and your account is hosted on one of them. I'll discuss how to find out which data center your account is attached to shortly.

<dc> is replaced by the string for your acconts data center. For example:

```
us1.api.mailchimp.com
```

<specific api>

The MC API is split into two pieces. The Export functions are grouped separately, because they typically involve large amounts of data, and the response is slightly different. The function for exporting a list is simply 'list':

```
list( string apikey, string id, string status,  
array segment, string since )
```

<parameters>

Despite the nomenclature of the API call above, the parameters are passed to the list api call are done so as you'd expect in a URL, with questions marks and ampersands separating them. The only parameters that are required are the apikey and the listid. So a URL call would, at the minimum, look like so:

```
?apikey=secret_apikey&id=secret_list_id
```

Put them all together and the API call would look like this:

```
https://us1.api.mailchimp.com/export/1.0/list?apikey=veryverysecret&id=alsoverysecret
```

Listing 1. A typical API key.

Finding Parameter Values

Theory is all well and good, but you're going to need the actual values for each of these parameters. Here's how to find them.

MC API keys

When you manually access MC via their website, you log in with a username and password. An API key is a string of characters that allows an application to access your MC account.

The short answer is to log in to your MC account, click on your profile name, click on 'Account Settings', and select the Extras menu, as shown in **Figure 1**.

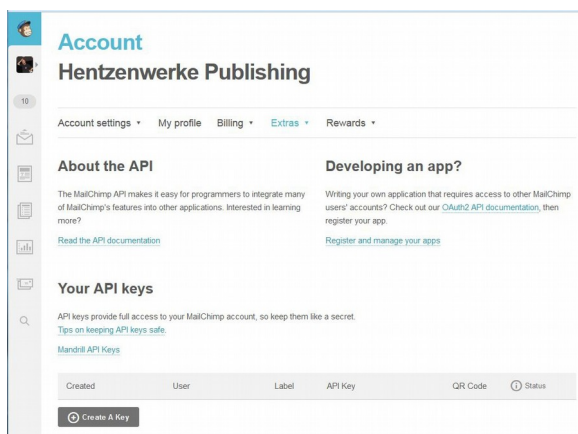


Figure 1. API key page.

Click on the Create A Key button in the lower left corner. You'll see a line displaying your key above the button, as shown in **Figure 2**.

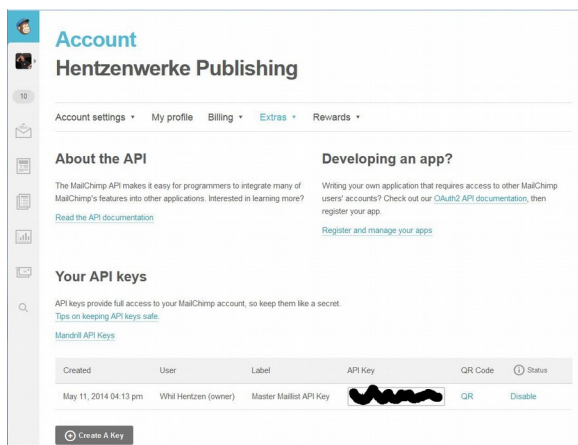


Figure 2. A generated API key.

Your API key is contained in a textbox that you can copy from and paste into your code.

If you need more details, the MC website has an excellent article that describes how to obtain an API key for your account:

```
http://kb.mailchimp.com/article/where-can-i-find-my-api-key/
```

MC List IDs

Your API key just gets you into the MC server. Since you can have more than one list, you need to be able to identify which list you are interested in. Each list has a unique ID. It's found by clicking on the Lists icon on the left (the icon with two pages), clicking on the name of the list you're interested in, clicking "Settings" in the middle of the page, and then on "List Settings and Defaults" in the Settings dropdown, as shown in **Figure 3**.

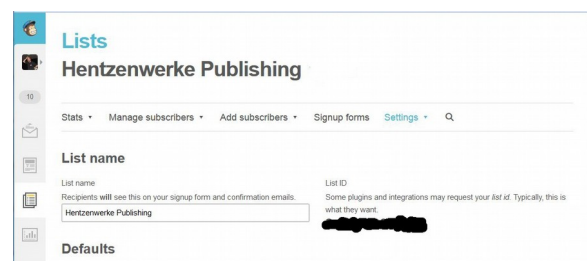


Figure 3. Finding your list ID.

Remember, each of your lists will have its own ID.

Finding your DC

Remember how I promised I'd tell you how to find your data center? It's contained in your API key, specifically, the last three characters. So if your API key looks like this:

```
js8e13wsf93jsnsf8sdhf-us2
```

your data center is 'us2'.

Testing your URL

It's easy to test that you've got all the pieces put together. Add your data center, api key and list id in the appropriate places, and you'll get a string much like that shown in Listing 1.

Paste it into your Web browser address bar, and after a few moments (depending on how big your list is), you'll get back a string in your browser that looks something like **Figure 4**.

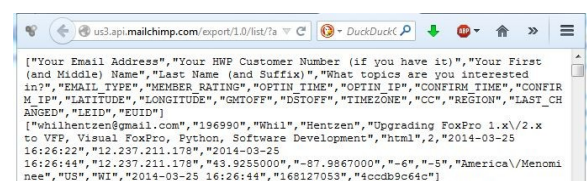


Figure 4. Results of calling the API manually.

Automating with the West-Wind Client Tools

So far, so good. But this isn't very automated. Let's look at using West Wind's Client Tools to make this happen automatically.

Installing West-Wind Client Tools

You really should have the West-Wind client tools as part of your arsenal already, but if you don't, you can follow along with a shareware version from

<http://www.west-wind.com/WestwindClientTools.aspx>

Click on the Download icon in the upper left, as shown in Figure 5.



Figure 5. Downloading the West Wind Client Tools

You can also simply buy it here:

<http://store.west-wind.com/product/WWCLIENT50>

You'll get a zip file named `wwclient.zip`. Unzip it into wherever you keep your third party development tools, creating a `wwclient` folder. For me, it's in my `'dev'` folder.

(By the way, don't you hate it when a zip file isn't clear if it contains a parent folder, so that you end up with a `'mynewdevtool\mynewdevtool\'` hierarchy? With `WWClient.zip`, unzipping via `"unzip here"` will create a bunch of files and folders, while extracting to `'wwclient'` will create a `wwclient` folder and the files and folders underneath.)

Configuring the West-Wind Tools

Next, include the following in your VFP path (substituting the fully qualified folders on your machine as appropriate):

```
\DEV\WWCLIENT; \DEV\WWCLIENT\CLASSES;  
\DEV\WWCLIENT\TOOLS
```

Now, start VFP and set the default to the folder that contains the source code for this article.

The conn2MCapi program

The main program, `conn2mcapi.prg`, connects to the MailChimp API and executes the functions we performed manually earlier in this article. It's pretty easy.

I think it makes sense to create variables for the data center, your API key, and your list ID, instead of creating a single URL that concatenates them. That way, this program will be reusable. I've used the same dummy values from earlier:

```
local lcDC, lcAPIkey, lcListID, lcURL  
lcDC = 'us2'  
lcAPIKey = 'secret_apikey'  
lcListID = 'secret_list_id'  
Then assemble the URL, like so:  
lcURL= "https://" ;  
+ lcDC ;  
+ ".api.mailchimp.com/export/1.0/list?apikey=" ;  
+ lcAPIkey ;  
+ "&id=" + lcListID
```

Next, load up the West Wind Client tools.

```
do wwclient  
do wwhttp
```

Instantiate the HTTP object.

```
o=createobject('wwHTTP')
```

Initialize the variable involved in moving data to and from the Mail Chimp website.

```
m.luResult = ''
```

And fetch the data, using HTTPGet.

```
m.luResult=o.HTTPGet(lcURL)
```

The `luResult` variable contains the data set fetched from the MailChimp API. It mimics the results displayed in Figure 4. It's simply a text string, each line separated by a carriage return and each field comma delimited.

```
["Your Email Address","Your HWP Customer  
Number (if you have it)","Your First (and  
Middle) Name","Last Name (and Suffix)","What  
topics are you interested  
in?","EMAIL_TYPE","MEMBER_RATING","OPTIN_TIME"  
,"OPTIN_IP","CONFIRM_TIME","CONFIRM_IP","LATIT  
UDE","LONGITUDE","GMTOFF","DSTOFF","TIMEZONE",  
"CC","REGION","LAST_CHANGED","LEID","EUID"]  
["whilhentzen@gmail.com","196990","Whil","Hent  
zen","Upgrading FoxPro 1.x\2.x to VFP, Visual  
FoxPro, Python, Software  
Development","html",2,"2014-03-25  
16:26:22","12.237.211.178","2014-03-25  
16:26:44","12.237.211.178","43.9255000","-  
87.9867000","-6","-  
5","America\Menominee","US","WI","2014-03-25  
16:26:44","168127053","4ccdb9c64c"]
```

The `wwHTTP` object has a number of methods and properties in addition to the `HTTPGet()` method. For our purposes, the `nError`, `cErrorMsg`, and `cResultCode` properties are useful if you're having trouble with the response.

Author Profile

Whil Hentzen is an independent software developer based in Milwaukee, Wisconsin (as opposed to Milwaukee, Minnesota, as many people think.) His writing has killed many trees over the years, but none since 2007. He has realized he really sort of misses it. You can reach him at whil@whilhentzen.com