

Solving an Email Send Problem: a Simple, Innovative Approach

Posted At : October 22, 2008 6:40 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Tips and Techniques

Getting email to go through for your customer or newsletter can be tricky these days. An email has to work pretty hard to make it through all the barriers in its way. The problem is compounded when you have an automated email system like a newsletter or marketing apparatus. Even if you follow all the rules you may still achieve a marginal success rate. One of our customers (Tom Long of digbro.com) has a unique system for marketing that is clearly *not* spam. It involves users sending individual emails out to customers with whom they already have a relationship. It's not bulk email and there are clear opt out instructions. The email goes out in a small trickle of a few hundred an hour. The system tracks the lead internally and an elaborate reporting system allows the user to figure out success rates, run referral programs and keep track of the customer information.

In spite of the benign nature of this system it has still been difficult getting emails out to some domains - particularly large email domains like Yahoo and Hotmail. We have done some innovative things like creating a "scoring" system that utilizes calls to spam assassin to try and "rate" the email before it is sent. The user submits his email text, the system dummies up an email, pipes it to Spam Assassin, retrieves it back and extracts the score - all before the email is sent. Pretty nifty - but not a panacea. In spite of our best efforts the success rate to known good email addresses on larger email domains improved only marginally. This customer is one of those visionary men that simply can't help thinking outside the box. Check out his solution.

Tom noted that naturally, emails sent from Outlook or Thunderbird tended to get through pretty well because the presence of "office like" or "client like" headers and other personal stuff tips the filters in favor of the email being from a person and not a system. Of course you can spoof headers but, besides being unethical, it is pretty easy to spot spoofed headers. The minute a filter flags your headers as spoofed it raises the spam score. So the trick is to get these emails to be created automatically, logged by the system and *still* get sent from the user's actual email client.

What Tom suggested was to allow the user to enter message information and user information as before, log the messages in our reporting system as before, but in the final step create a *mailto:* link that pops up in outlook with the requisite message and subject line (and tracking link). This seemed deceptively simple, but it worked really well. You probably already know that mailto links can contain subject and body attributes that end up in the subject and body of the message. Like so:

```
<a href="mailto:joe@thecrabshackaroo.com?subject=Craborama&body=Hello World">email joe</a>
```

The above link looks like this: **email joe**. If you click on it it should open up your client email with a subject of "Craborama" and a body of "hello world". So all Tom's users had to do was generate a page of links and tediously click on them one at a time and send. Hmmmm.... we needed a way to make it slightly more user friendly. Enter the "iframe".

```
<div id="hiddenDiv">
<Cfloop query="emailstosend">
  <iframe
    style="height:0px; width:0px;"
    src ="mailto:#email#?subject=#subject#&body=#urlencodedformat(body) #" >
```

```
</iframe>  
</CFLOOP>  
</div>
```

This rather handy piece of code is hidden on the page. When the page loads it pops up a "new message" window for the email client containing the message that is to be sent. Multiple windows pop up one on top of each other and the client can send each of them quickly using control+s or clicking "send" in the same spot on the screen. This is about as streamlined as we could make it while still utilizing the users email client.

The downside

Of course this doesn't solve Tom's issues completely. He has traded a fully automated system for one with more human intervention - which is exactly why it has a shot at working with anti-spam systems. Still that means he is dependent on the SMTP configuration of hundreds of email clients that are out of his control. He will have users with dozens of different clients and versions of clients. His users with web based email are out of luck and he cannot predict what restrictions will be placed on future browser behavior or email client behavior. Finally the "body" parameter can only carry plain text messages (no HTML). So in the end he has restricted his users to a fairly narrow set of programs and parameters. Still, for what's it's worth, it is not really *much* more difficult than the fully automated system, and recipients downstream are getting a bona fide personal email. I'm anxious to see how it improves his send rate.

Meanwhile, some spam jockey who is also an avid muse reader will no doubt set me straight on some of the alternatives. I look forward to those comments as they add to my burgeoning arsenal of tools - so post away (just be nice:).