Designing, Coding & Delivering
HTML Email

···

A beginners survival guide from
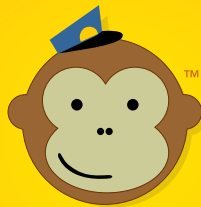
MailChimp

Table of Contents

Introduction

You don't have to be a professional web designer to use this guide, but a little HTML knowledge will definitely help. First, we'll cover all the **basics**, like how HTML emails work (and why it always seems to break when you try to send it yourself). Then, we'll get into **technical stuff**, like how to design and code them. Finally, we'll run through email marketing **best practices**, such as for list management, deliverability and measuring performance.

About the Third Edition

According to our server logs, this guide has been downloaded over 100,000 times since we first wrote it in 2001. Hopefully, it's helped a few people out there get their email marketing off to a good start.

We wrote the first edition because when we started MailChimp, you had to know a lot of technical stuff before you could get into email marketing. Back then, an email marketer was a jack of all trades, designing their emails, coding them into HTML, setting tracking links and images, managing bounces, and setting up databases on their websites.

Nowadays, email marketing services (like MailChimp) have launched tools that do all that mundane stuff for you. We manage your lists, give you signup forms, send welcome emails, handle double-opt-ins, set one-click unsubscribe links, and we even give you built-in HTML email templates (so you don't even have to know how to code HTML email anymore). All you have to do now is come up with good content, and click send.

But there are new challenges, which we'll cover in this 3rd edition:

- Most email programs now block all images in your HTML email by default. You've probably noticed how you have to right-click on an email, or push some button, to make images show display in the emails you receive. How should you design around that?
- Desktop email programs (like Outlook, Lotus, Eudora, and Apple Mail) have made great progress in standardizing the way they display HTML email. But now webmail services (like Yahoo, Gmail, and Hotmail) are getting more popular, and they have their own unique challenges. Two steps forward, one step back. What works and what doesn't in webmail clients?

- Spam filters are very sensitive to the content in your email. How should you write your content?

- There's so much spam out there, spam filters need their own spam filters now. They're called firewalls, and they block email before they even get to your recipients' spam filters, based on reputation. How do you protect your reputation?

- Spam laws require that every email marketer follow some very important rules now. If you break any of the rules, you can get your pants sued off. What do you need to include in every single email you send to your customers?

- Blacklists used to simply block email based on what server you sent from. Now, they scan the content of your messages, and they look for domain names that have been found in reported spam. Even if you've never sent an email campaign before, you can find yourself on a blacklist, if one of your reseller partners has been sending spam with your domain name in it.

But First: "What The Heck is MailChimp?"

MailChimp is a product of The Rocket Science Group, a web development company in Atlanta, Georgia. Our one, driving motivation is to "build tools that are **U**seful, **S**imple, **E**fficient, and that people *love* to use." **U.S.E.** is the formula and philosophy for everything we do (incidentally, it started out as s.u.e, but that sounded way too litigious).

Way back in 2001, we noticed some of our clients were trying to send out HTML email campaigns and monthly newsletters to their customer lists, but stuff kept breaking. They were rigging Microsoft Outlook "stationery", or trying to send ginormous PDF attachments. Images were missing, tables were broken, formatting was off. Their customers were complaining. They were exceeding their ISP bandwidth quotas. Bounces were out of control. Sheesh! It was a *mess*.

They could handle *design* just fine. Even a little HTML coding. But it was the ***delivery*** part that they needed help with. See, to send HTML email *properly*, you have to set up a server to deliver in "Multipart-Alternative MIME format," then track bounces with "VERPs" and other geeky stuff like that (we'll go over that soon).

So we built a simple little tool where they could log in, copy paste their customer list, copy paste their email code, and hit "send." And since *they* took care of the *creative* stuff, and we just handled the *boring* tasks that "only a helper chimp should do," we called it "MailChimp." We've since added some "email design" tools to MailChimp, to help people who don't know one bit of HTML. Now, you don't have to be a web designer (or even have access to your own server) to use MailChimp.

Thousands of customers from all over the world have discovered the simplicity of MailChimp. Maybe if you find this guide useful, you'll give MailChimp a try, too (trust us, you'll love it).

Learn more at: http://www.mailchimp.com/

How HTML Email Works

Before you can start designing, coding, and sending HTML emails, you should know how it works and what tools you'll need. Here's some background information every email designer and marketer should know…

The Multipart/Alternative MIME Format

The most important thing you need to know about HTML email is that you **can't just attach** an HTML file and a bunch of images to a message and hit "send." Most of the time, your recipients' email applications will break all the paths to your image files (because they'll move your images into temporary folders on your hard drive). And you can't just paste all your code into your email application, either. Most email apps send messages in "plain-text" format by default, so the HTML won't render. Your recipients would just see all that raw source code, instead of the pretty email it's supposed to render.

You need to send HTML email from your server in "**Multipart-Alternative MIME format.**" Basically, that means your mail transfer agent bundles your HTML *code*, PLUS a plain-text version of the message, **together into one email**. That way, if a recipient can't view your beautiful HTML email, the good-old-fashioned plain-text version of your message is auto-magically displayed. It's kind of a nerdy gobbledy-geek thing, which is why a lot of people mess things up when they try to send HTML email themselves. You either need to program a script to send email in multipart/alternative MIME format, or just use an outside vendor (ahem, like [MailChimp](http://www.MailChimp.com)) to deliver things for you.

Image Files in HTML Email

Embedding images and photos into messages is the number one reason people want to send HTML email. The proper way to handle images in HTML email is to *host them on a web server*, then "*pull them in*" to your HTML email, using "<u>absolute paths</u>" in your code. Basically, you can't send the graphics along with your message. You host the graphics on a web server, and then the code in your HTML email downloads them whenever the message is opened.

Incidentally, this is how "open tracking" works. You place a tiny, invisible graphic into the email, and then track when it's downloaded. This is why open tracking only works in

HTML email, not plain-text, and why the new email applications that block images by default (to protect your privacy) can screw up your open rate stats.

<div style="border:1px dashed; background:#ffffcc; padding:1em;">

When coding Image Tags in HTML email...

Do this:
<img src="http://www.yourserver.com/email/images/logo.gif">

Instead of this:
<img src="images/logo.gif">

Note: If you just use MailChimp's built-in email designer, we host your graphics on our server for you (free). No need to FTP files anywhere, or code anything.

</div>

Free Hosting Services

When it comes to hosting the images for your HTML email, you really need your own server to do it. Don't try hosting images on a free "image hosting service," because those websites often put scripts in place to prevent you from linking to them in emails (they can't handle all the traffic). And since you really do get what you pay for, free image hosting services tend to be pretty unreliable under heavy traffic conditions. Also, spammers use free image hosting services all the time, to "cover their tracks." If you don't want to look like a spammer, use your own web server. If you use an email marketing service (like MailChimp) they usually come with a newsletter builder tool with image hosting capabilities built-in.

Delivering HTML Email

Many newbies make the mistake of setting up forwarding lists, or "CC'ing" copies of a message to all their customers. This causes all sorts of problems, like when a customer hits, "reply to all." Plus, there's no way to do any kind of individual tracking or personalization when they CC: a big group like that. Finally, it just looks so unprofessional and impersonal when recipients can see your entire list of other recipients like that. Blech.

That's why when an email marketing system (like MailChimp) sends your campaign, we take your message and send it one at a time to each recipient on your list (really, really fast). Unlike your work computer linked to your local ISP (which probably has a standard monthly bandwidth limit), email marketing vendors like us use dedicated mail servers that are capable of sending hundreds of thousands of emails (even millions, for larger vendors) per hour.

Other factors you should consider when delivering HTML email campaigns:

- If you send from your own server, your ISP may throttle your outgoing emails or shut your account down, if you send too much too fast (like more than 500 per hour). They may also shut you down if you exceed your monthly bandwidth limit. If you've ever sent an email campaign to a big list of recipients, and you attached a giant PDF, this has probably happened to you before.
- Email firewalls and ISPs who receive your emails don't like receiving tons of emails from one single IP address at once (unless they *normally* receive tons of emails from that IP). So if you only send occasional email campaigns from your IP, you may want to throttle your delivery, or spread them across multiple IPs, to avoid accidental blocking. Email marketing services usually split your campaign up into pieces and send it out over lots of different IP addresses.
- If you send emails from your desktop email program, chances are you're connecting through your local ISP. If you don't have a dedicated IP address setup with your ISP, you're probably sending emails from a dynamic IP address. ISPs and spam filters don't like receiving lots of emails from a dynamic IP address, because it looks like a hijacked home computer. You should always send from a dedicated IP address, or use an email marketing vendor (like MailChimp).

Designing and Coding

Now that you have a basic understanding of how HTML email campaigns work, and how you should deliver them, we can start talking about designing and coding them (the fun part!).

Tools of the Trade

If you want to get good at coding HTML email, here are the tools you'll need...

1. **A design application**, like Adobe Photoshop, Fireworks, Dreamweaver, or Illustrator. Use these tools to layout your template, and slice out your graphics (like your company logo, and product photos). We're really partial to Fireworks, but to each his own.

2. **A good, "pure" HTML/Text editor**, like BBEdit and TextWrangler for the Mac, or HomeSite, and NoteTab Pro for the PC. You *could* use "WYSIWYG" style tools to generate HTML, like Microsoft FrontPage, or maybe Adobe GoLive or Macromedia Dreamweaver. But they have their disadvantages. WYSIWYGs very often throw in gobs of junk into your code (FYI, Microsoft says the extra code is to help with "round tripping" which means you can export from Word to Powerpoint to HTML, then back again, without losing anything). All this extra code usually ends up just breaking things, or setting off spam filters. There are some WYSIWYGs out there that generate "clean code" but we've found those are a bit "*too perfect,*" because they're designed for web pages, not HTML email (so they don't know all the things you have to "rig" in your code.). If you want to do HTML email *correctly*, you should learn to code HTML "from scratch" with a good text editor.

> ## Free HTML Email Templates:
>
> If all this talk of designing and coding HTML is too nerdy for you, consider using MailChimp's built-in HTML email templates. They're super flexible, and can be customized to match any brand. Plus, they've already been tested in all the major email programs. See some nice examples of what our customers have done:
> http://www.mailchimp.com/screenshots/template_examples/
> and watch a demo movie of our template design tool:

3. **Your own web server**, to host all your images and archives. You should create a folder on your website for email newsletters. This is where you'll store images, then point to them with absolute paths in your HTML code. You might also want a folder on your website to store archived newsletters. When we first started out, we created a folder on our website called, /monkeywrench/ and then kept campaigns under subfolders named by /date/ to store each campaign. Everybody has their own organization style, though. Just be sure to come up with a system and process for archiving, so that each newsletter isn't a huge chore to publish.

   Nowadays, we don't even bother, because we use MailChimp's built-in archive functionality. It'll save a copy of every campaign you send from our system. You can build a page on your website called "Newsletter Archives" and paste a little code snippet from MailChimp. We'll keep your page updated with all your most recent campaigns. Saves so much time.
4. **An FTP program,** like Fetch for the Mac, or CuteFTP for PCs. When you design your email and slice out all your graphics, you'll use FTP to move those files from your computer to the server.
5. **A test machine (or two)**, loaded with as many different email applications and accounts as possible. You'll want to check your HTML email designs under lots of different scenarios. Unlike web design, where you only have to check your work in a small handful of browsers, there are *tons* of email applications out there, and they all handle HTML email differently. If you use MailChimp, you can use our Inbox Inspector tool and let us do all the testing for you: http://www.mailchimp.com/add-ons/inboxinspector/

Design Guidelines

Here are some pointers for designing your email. Early warning: If you're a seasoned web designer, you're not going to be happy with all the compromises and hacks you'll have to make when designing for email applications.

Set your width to about 500-600 pixels.

If you've ever worked on a website, you're probably used to designing pages to fit in 800x600 pixel resolution screens, or maybe even 1024x768. But that won't work when you design HTML email. Most recipients will be looking at your email through their "preview pane" which is usually a small portion of their available screen (see screenshot below).
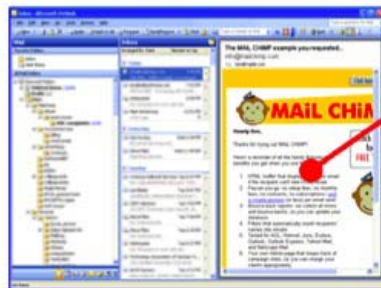
Above is a screenshot of an HTML email in Microsoft Outlook's preview pane. Outlook is maximized to full screen (1024x768 pixels), but as you can see, the preview pane is only about 440 pixels wide. In **AOL 9**, the "preview pane" is only about 194 pixels wide, then when you click "Full View," it typically opens up a window that's about **540 pixels** or so. This is why you shouldn't design emails to be much larger than **600 pixels** in width.
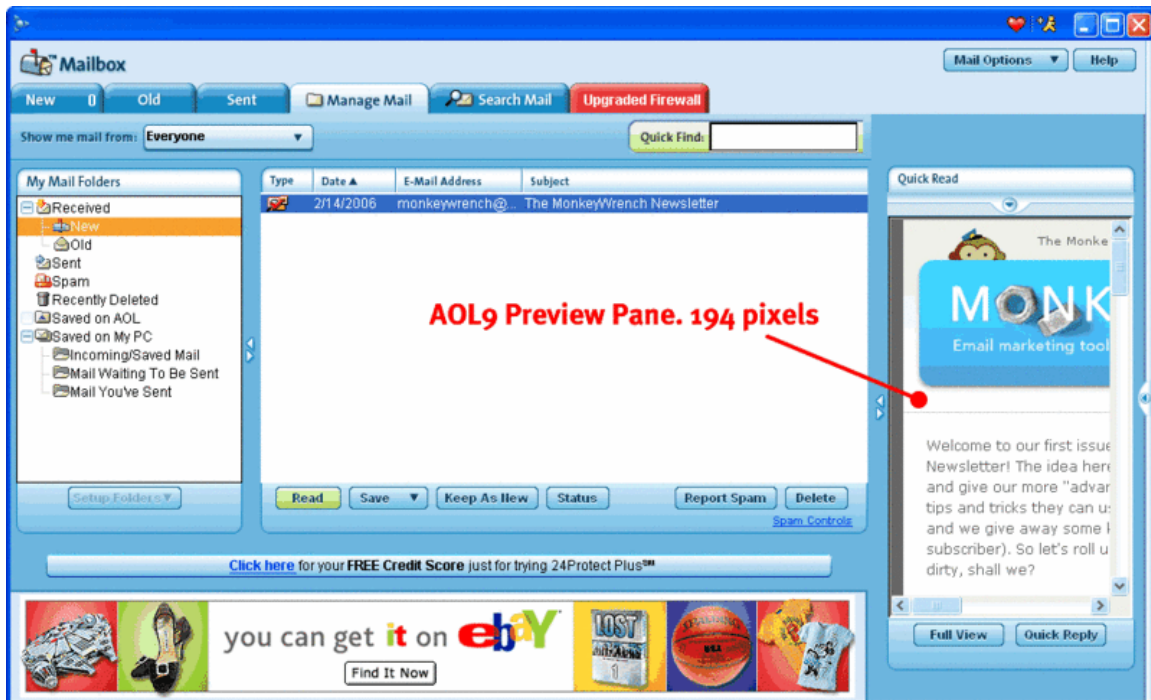
Simple Layouts and Tables Work Best

You wouldn't believe all the different ways email applications garble up HTML email. You'll need to keep your email designs nice and simple. Avoid complicated layouts, too many embedded tables, and tables with too many rows and columns. A simple two column table with a row across the top is about as complex as you can get. There are lots of email applications that don't cooperate when you code tables with colspans (table cells that "span" across multiple columns) in them. One application that's particularly finicky when it comes to tables (and just about everything *else*) is Lotus Notes. We can't stress it enough. You've got to keep your layouts very, very simple, if you want it to work across all the major email apps.

If you're working on an email layout and find the tables are getting complex, and you're having to code too many "COLSPANS," you might think about chopping it up into separate tables. For instance, we almost always use separate tables for the header, body, then footer of our emails. Better to go with more "simple" tables, than one big "complex" table.

If you're a CSS ninja, you're probably rolling your eyes in disdain with all this "TABLES" talk. But trust us --- you're not going to be able to use all that cool, timesaving CSS stuff

in your HTML emails. DIVs and CSS-positioning don't work. We'll cover that later in this guide.

Special Considerations for Webmail Services

A lot of your recipients will be checking their email in their **web browser**, such as with Yahoo!Mail, Hotmail, or Gmail. Since they're viewing email in browsers, there are certain things you need to know when you code your HTML emails:

- They'll strip out your <HTML>, <HEAD>, and <BODY> tags, in order to keep *your* code from interfering with *their* web pages. That means...
- Any background colors you specify in your <BODY> tag will be lost. So wrap your entire email **inside** a larger, 100% wide table, and set a background color to that table.
- Any CSS that you place inside the <HEAD> tags will be lost. So don't try to link to CSS files hosted on your server. Use embedded CSS instead, and make sure it's below the <BODY> tag. If you're a web designer, that'll definitely make you feel "kinda dirty" because that's not exactly the "proper" place to put embedded CSS. But no worries---it'll work. If you just can't stomach that, use inline CSS.
- Some webmail clients are starting to strip even the most basic CSS, even if it's embedded below the BODY tag (like we described above). We think it's because they don't want any of *your* CSS interfering with *their* CSS, or maybe because they're using so much JavaScript and AJAX technology, stuff just gets buggy. Keep that in mind, and do lots of testing
- **Tip:** you can use MailChimp's Inbox Inspector to get screenshots of your work in all the major email programs and webmail clients, to see if your CSS is breaking. Saves a lot of time and hassle.
  http://www.mailchimp.com/add-ons/inboxinspector/

CSS in HTML email

CSS (Cascading Style Sheets) is a Godsend for websites, but don't put too much faith in it with HTML email. For instance, you shouldn't expect DIVs to work. And *definitely* stay away from CSS *positioning*. Won't work. If you're a CSS-standards freak, you are going to hate yourself after coding HTML email.

You'll have to rely on old-fashioned <TABLES> for your layouts in HTML email, and only use CSS for simple font formatting and colors. Always design your CSS to "fail

gracefully." That means if someone took away your CSS, your design and content would still display decently. Before you send your HTML email, delete the CSS and see what it looks like. Did you have tiny text that is now gigantic, and blowing out your tables?

CSS is *especially* a touchy situation if you're using a WYSIWYG to design your HTML email, because WYSIWYGs insert all kinds of crazy CSS and DIVs by default. It's one of the drawbacks of WYSIWYGs for HTML email. It may help you get things coded, but you still have to understand HTML enough to go back and *remove* some of the code, so things won't break in email applications. You might as well code by hand.

Using Microsoft Word to generate HTML is even worse than using WYSIWYGs. Word adds so much code behind the scenes, it's unbelievable. Don't do it. Learn to code HTML, or use our built-in HTML email designer.

If your email design uses lots of CSS, be sure to check it in Google's Gmail. At the time of this writing, they don't fully support CSS, so you'll see your fonts defaulting to Arial, and black in color. Here's what our MonkeyWrench Newsletter looked like in Outlook 2003 (which supports CSS just fine) vs. Gmail...

Gmail stripped out virtually all of my CSS! Where did my font sizes, font colors, and line-height go? And since the fonts are so much larger in Gmail, it could have blown out my table cells (if I didn't know what I was doing). But notice the top line (next to the peeking chimp). It wrapped my text into 2 rows. Blech!

Flash, (and JavaScript, ActiveX, movies, and other stuff that won't work) in HTML email.

You can *deliver* HTML emails with Flash in them, but most recipients won't be able to *view* them. Not unless they're using an email program that uses a browser to render their HTML email. Microsoft Outlook was a program that used Internet Explorer to render HTML email, so Flash would sometimes work there. But Microsoft Outlook 2007 is no longer going to use IE to render email. They're using Microsoft Word instead, so Flash is not going to work there anymore. Anyways, most **anti-virus applications** block the code used to embed Flash movies, and they block JavaScript, ActiveX, and even background music files. All that stuff has been used in the past to spread viruses, so it's commonly blocked now.

Point emails to landing pages

If you've got a great animation or movie to show to your recipients, just send a simple, intriguing GIF or JPEG graphic in your HTML email, then link it to a "landing page" on your website with the animation in it. Same goes for JavaScript, ActiveX, and movie files. Anti-virus applications block them from running. So fancy rollover or pop-up navigation and streaming videos just aren't going to work either. The only email application on our test machines that will actually play fancy stuff (like Flash and movies) seems to be Outlook 2000 (which is kind of old, these days). Unless you know for a fact that every single one of your recipients uses Outlook 2000, and you know their anti-virus applications won't block your stuff from running, don't send Flash, JavaScript, ActiveX, or movies in HTML email. Just won't work reliably.

Background Colors

Remember when we told you (above) about how browser-based email services (like Yahoo!Mail and Hotmail) strip out your <HTML>, <HEAD>, and <BODY> tags? That's what you need to remember when coding your background colors and images. You

normally specify that stuff in your <BODY> tag, but you can't rely on that with HTML email. You'll need to create a big, 100% wide "table wrap" around your email. Set your background color in that table, and use some CELLPADDING as needed. If you're into CSS, you might be tempted to use a <DIV> instead. But we've seen DIVs break quite frequently in email applications, so we don't recommend them.

Anatomy of a Good, Healthy HTML Email Newsletter

Here's what a well-built HTML Email Newsletter looks like:
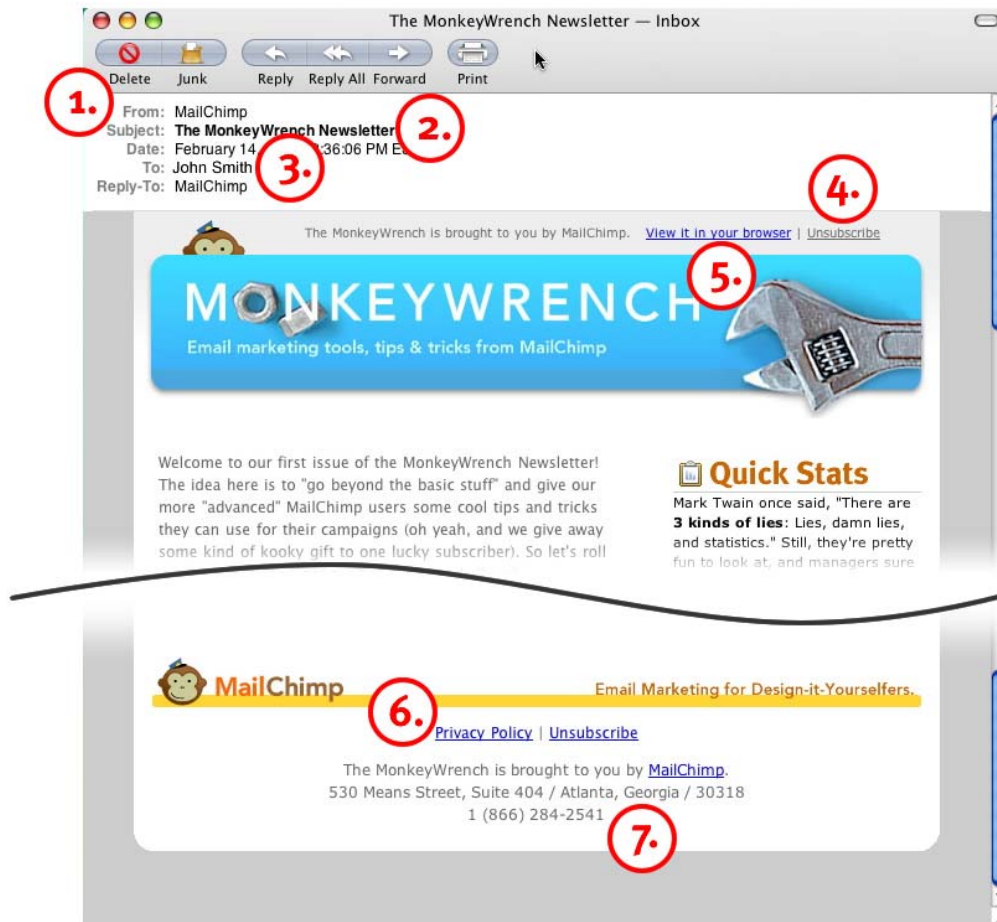
1. Your company name in the "From." Recipients should recognize who the email is from <u>instantly</u>. It can't be deceptive in any way (duh). If a recipient has to strain his brain to remember who you are, he'll click "this is spam" instead of opening.
2. A relevant subject line (don't be "spammy"), with your company or newsletter name in it. So they instantly know who the email is from, and what it's about (hence, "subject" line).
3. The "To:" field of your email should be personalized to the recipient's name, not their email address. MailChimp lets you "merge" recipient names into this field, if you have it in your database. You would insert *|FNAME|* *|LNAME|*, for

example. Search our help documents for "merge tags" if you want detailed instructions.

4. A one-click opt-out link that removes people from your list immediately. Consider placing it at the *top* of your email (as well as in the footer), so that people who want off your list can find it really easy (instead of clicking their "Junk" button). If you use MailChimp's Managed List functionality, you would use our [*|UNSUB|*](#) tag to generate your unsubscribe link (it's required in every email you send from our system). Any recipient who clicks it will be instantly removed from your list. If you use our built-in HTML email templates, we insert that link into the footer for you (no coding required).

5. In addition to your opt-out link, you might also include a link in your header for recipients to "view this email in your browser." Point it to an archived version of your email on your server. This helps if the email got forwarded to friends, and then got mangled along the way. MailChimp automatically keeps an archived version of all your campaigns on our server. You can just use our *|ARCHIVE|* tag in your emails to link to yours. If you use our built-in HTML email templates, we automatically place this link at the top of all your emails.

6. A link to your company's privacy policy should be in your email. If you don't have a privacy policy, you can use the DMA's privacy policy generator. Google the phrase, "DMA privacy policy generator"

7. Your valid, physical mailing address (P.O. Boxes aren't good enough), and as much contact information as possible. The more contact information you provide, the more reputable your email will look.

8. Bonus: It's a really good idea to also include some kind of reminder text, like "You are receiving this email because you signed up at our website." People forget opting-in to lists, and they get a little trigger-happy with the "this is spam" button. That can get you reported to the major ISPs, so you want to prevent that as much as possible. Also, in the rare case that a recipient reports you to an anti-spam organization, having this reminder text can make the difference between a server admin blacklisting you forever, or contacting you for further explanation. If you use our built-in email templates, MailChimp inserts that reminder text into your email footer automatically.

Plain-text Email

Don't forget your plain-text email! You need it for people who can't (or won't) view HTML. Sometimes, the plain-text version of your email is the one that gets displayed if your recipient checks his email on his mobile device.

If you don't take the time to create the plain-text version of your HTML email, you'll just look like a lazy spammer (spam filters will penalize you for only sending HTML). When you send a campaign in MailChimp, you'll need to create both versions of your message (HTML & Plain-text). Then we "bundle" them together in "multipart format" and send them off. Your recipients' email applications then decide which version to display.

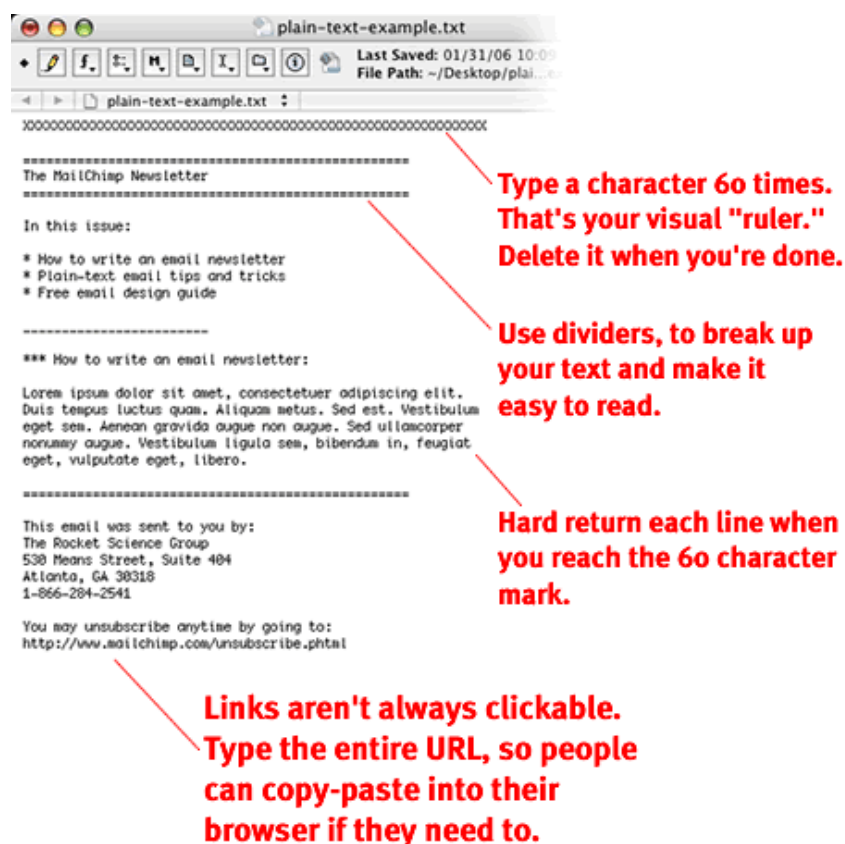### The Art of Plain-text email

There's definitely an art to constructing your plain-text email. You can't just use the same text from your HTML email. It has to be formatted *juuuust* right, so the best way to do it is to build it from scratch, or copy-paste the content from your HTML email, then reformat it heavily.

Here are some tips for your plain-text emails:
1. Use a plain text editor, like TextWrangler for the Mac, or Windows NotePad. Don't use something like Microsoft Word, because it won't give you the kind of "raw" text files you need. You need an application that gives no formatting whatsoever.
2. If you're using MailChimp, create a plain-text message in your text editor, then simply copy-paste it into the "plain text message" field in MailChimp. No uploading required. Or, you can just type straight into MailChimp's interface. One thing I like to do is pop-up preview my HTML email, highlight all my content, then paste it into the plain-text field in MailChimp. Then, I go and reformat everything.
3. By default, some email applications start to wrap your plain-text messages at about 60 characters or so. That's not really a big deal to most people. Just let your text wrap wherever their email programs want it to. But if you have content that you absolutely do not want to wrap unexpectedly (like in the middle of a hyperlink), one nice little trick is to type a letter 60 times across the top of your message, and use that as your visual "ruler." As your content approaches the end of that ruler, hit your return key and "hard wrap" the line. Again, this is only if

you want very strict formatting. These days, I let my plain-text emails just wrap fluidly.

4. People don't read email. They *scan* email. Especially plain-text email. So make it really easy to find important information with bullet points and visual markers (see screenshot below)

5. You can't code a clickable link with plain-text email. You just have to type out the entire URL. Some email applications will make it clickable automatically. Some won't.

One thing we've noticed is that a lot of people spend so much of their time on the HTML version of their message, they're too "drained" to work on their plain-text email (we're guilty of that ourselves). So they either ignore it, or they slap some junk in, or they "cheat" a little, and type in something like, "Evidently, you can't view HTML email, so visit this URL in your browser to see our newsletter in all its glory." We think this is a big mistake. So don't put *all* your energy into the HTML version of your email. Save some love for your plain-text message too. If you find yourself spending so much time coding

HTML that you're too pooped to make a plain-text email, you should consider setting up one re-usable template that you can just select, and "fill with content" every campaign. No need to reinvent the wheel for every campaign.

MailChimp Trick:

When you setup a managed list in MailChimp, you have the option to "let my users choose plain-text only" when they fill in your opt-in form. MailChimp will automatically deliver *only* the plain-text formatted email to them when you send your campaigns.

MonkeyBrains!

MailChimp's MonkeyBrains blog is chock full of HTML email design examples, tips, tricks, and hacks, plus email marketing news and best practices.

http://mailchimp.blogs.com

Common Mistakes to Avoid

Here are some of the most common mistakes people make when coding HTML emails:

1. Not coding "absolute paths" to their images. Remember, attaching graphics and using "relative paths" won't work. You need to host the images on your server, then link to them in your code.

2. Using JavaScript, or ActiveX, or embedding movies. That stuff just doesn't work in HTML email.

3. Getting over ambitious with designs. Designing HTML email isn't the same as designing web pages. You can't have all the CSS-positioning, DIVs, DHTML, and complex, embedded tables like you can in web pages.

4. Forgetting---or refusing---to include an opt-out link. It's stupid and unprofessional not to allow recipients to unsubscribe from your list. Oh yeah, it's illegal, too. Be sure to read and understand The CAN-SPAM Act of 2003.

5. Linking to an external CSS file. You usually put CSS code in between the <HEAD></HEAD> tags of web pages, right? But browser based email applications (like Yahoo, Hotmail, and Gmail) strip <HEAD> and <BODY> tags from HTML email. Embed your CSS below the <BODY> tag.

6. Letting your permission "grow cold." If you've been collecting opt-ins at your website like a good email marketer, but you haven't sent an email in several months, your subscribers have forgotten all about you. So when they receive a full blown email newsletter from you "out of the blue" they're going to click their "this is spam button" in their email application (related blog post: Surviving the inbox whack-a-mole game). Don't let your permission grow cold. Send occasional emails. If you've been collecting emails for years, and you're only *just now* sending your first email to everyone, you should send a little "re-invitation" to your list. Something like, "A really long time ago, you subscribed to my email list. Well, I've finally got some time to start sending my really cool newsletter, and I want to make sure you still want to receive it. Click here to sign up..." At the very least, send an email and place some kind of text at the top that reminds them of where and when they opted in. People usually forget about opting-in to something after about 6 months or so.

7. Sending to a list without permission. This is the worst offense. Lots of "innocent" marketers, who "mean well" commit this heinous crime. Here are some common ways legitimate marketers can inadvertently become known as "evil spammers:"

- Getting an email list from a tradeshow. "But I'm exhibiting there, and the tradeshow host said it would be okay, and when people purchased tickets, the fine print said that we could email them, and..." Nope. You're spamming. Don't do it. If they didn't give you permission to email them, they didn't opt-in to your list. If they didn't opt-in, you're spamming. Even if you can legally send them email marketing, those recipients are more likely to report you for spamming them. Then, you'll get blacklisted. Plus, your company will look really slimy. Don't do it. If a tradeshow host is collecting email addresses, then they should be doing the emailing. It's all about permission, and setting expectations.
- Getting a list of "fellow members" from some trade organization. Just because they joined a club, and the club posted contact information so everyone could keep in touch with each other, it doesn't mean each member gives you permission to send them newsletters and offers. If you sent them a personal greeting from your own email account, they probably wouldn't mind. That's what the organization's "members list" is for. But add them to a list and send them a huge email newsletter, and you'll be reported for spamming. Don't do it.
- They go to events, and swap business cards. The business cards just get thrown into a "prospects" pile. Years later, they get an intern to finally type all the contact info from those cards into a database. Then, one day out of the blue, they send a big, fat email newsletter to everyone. As if they actually want to hear from you! Trust us, they don't.
- Fish bowl of business cards. Similar to the example above, but they hold an event, and collect business cards in a fishbowl to win a "door prize." The people who dropped their business cards into the bowl wanted the coffee t-shirt you're giving away (plus all the fame & glory). They didn't opt-in to your list.

8. Using a WYSIWYG to "code" your HTML. WYSIWYGs are notorious for generating absolutely horrible HTML. They insert so much junk code, it's unbelievable. Even the ones that generate "clean" code don't know how to "rig" things to work in email applications (like sticking your embedded CSS *below* the <BODY> tag). To code HTML email properly, you need to learn a little HTML. It's really not that hard. Plus, you can download free HTML email templates at our website: http://www.mailchimp.com/resources/templates/

9. Forgetting to test. Thoroughly. When you send HTML email, you've really got to test it in as many email applications you can. Then you have to test on different operating systems. Then different ISPs (we'll explain shortly). If you keep things simple, and build a rock solid, thoroughly tested **template** for each newsletter, you won't have to test so much. But you should *always* send at least a few campaigns to yourself before sending it out to your entire list.
10. Sending nothing but a big, gigantic graphic as their HTML email. Sigh, that's what spammers do. And since most email programs block images by default, what do you think your recipients see when (if) they open your email? Many spam filters will block your email if you don't have a healthy balance of images *and* text.

The MonkeyWrench Newsletter

Subscribe to MailChimp's MonkeyWrench Email Newsletter. We'll keep you up-to-date on more "advanced" email coding tricks, obscure hacks, power tips from MailChimp programmers, and oh yeah---free prizes.

http://www.mailchimp.com/monkeywrench/

Designing around spam filters

A guide on email newsletter design wouldn't be complete without a section on spam filters. You have to design your emails a certain way, if you want your messages to make it past them. But you can't go overboard, either. Spam filters can smell fear. If you try too hard, they know it, and they attack. So just be cool, act normal, and don't make any obvious mistakes.

How anti-spam systems work

Before getting into the nitty-gritty design tips, you need to know how all the various anti-spam mechanisms work out there. Once you get an overall understanding of how they work, designing around them will be much easier.

Bayesian Filtering

This is one of the most important ones to learn about, since it's installed in so many email applications these days (like Outlook, Mozilla Thunderbird, and Apple Mail). Bayesian filters work by watching users classify email as "junk" (such as when they click a "this is spam" button). It reads the junk mail, compares it to other emails you called "junk," and looks for common traits in the subject line, the content, the hyperlinks, the sender, etc. Over time, Bayesian filters learn to scan for those "traits" in every email message you receive. Every time they find something that looks "spammy," they assign a "score" to it. For instance, using "Click here!" might get you 0.7 points. Using **bright red fonts** might get you 2 points. Including the word, "mortgage" might get you 1 point, but using the word, "Viagra" will get you 5 points. Once an email exceeds some threshold (set by the user), the email is classified as spam, and thrown into the junk folder. It's amazing how many different things Bayesian filters look for. To read some, go to the MailChimp Blog and search for "Funny Spam Assassin Criteria"

MailChimp also comes with a spam filter checker (look for the "Inbox Inspector" under our Account tab). It'll tell you exactly what your email's spam score is, and what you need to change if you want to improve your chances with spam filters.

Black Lists

A while back, some server admins got really, really, angry at all the junk mail they received. So they started to track the IP addresses of the servers that sent them spam, and put them on "blacklists." Anytime they received email from a server on their blacklist, it was deleted immediately. It worked pretty well. So they started to share their lists with other server admins. And other admins started to add on to those blacklists. Pretty soon, the blacklists got very, very big. Large ISPs started to sync up with them. If you send email that's very "spammy" you could end up on one of these blacklists (whether the recipients are opt-in or not). Once you make it on to a blacklist, good luck getting off. Geeks are notoriously difficult to negotiate with (we know, because we're geeks). This is why you need to make sure you only send to recipients who gave you verifiable permission to email them, and you need to have _proof_ that each one of them opted in to your list (such as through a double opt-in system). Want to know the easiest way to get blacklisted? Let your company's sales team blast out an email newsletter to a list of "prospects" that they collected from conferences, and that they scraped off of websites. You'll get blacklisted, guaranteed.

Email Firewalls

Managing email servers (and incoming spam, viruses, and phishing attacks) can be a lot of work. So large corporations usually install "email firewalls" (Google the terms, "Barracuda Firewall" or "Postini" for some examples) to handle their incoming email.

Think of them as "spam filters on top of spam filters." They're big, heavy-duty gatekeepers, and they're not friendly at all.

They often use a combination of Bayesian-style/adaptive filters, community reporting, blackhole lists, and a little bit of proprietary "magic pixie dust" to keep spam out of the company. Most of the time, when your email's not getting through to a larger company, it's their firewall. You can think of these firewalls as kind of Xenophobic and paranoid. They're all twitchy, and tend to ask questions like, "_Okay, is this sender new to me? Why is he sending copies of the same, exact email, to a bunch of people in our company? Spammers do that kinda stuff. Hmm, how long has their server been around? Can I really trust this sender?_" Spam firewalls are usually only a problem when you first start sending campaigns to a big client or something. You'll experience some deliverability issues in the beginning, because you're "new." They'll eventually "learn" to let you

through. To expedite things, you may have to ask the IT people in charge of the firewall to "white list" your IP Addresses (or the IP address of your email service provider).

If you're interested in learning more about email firewalls, search for "email firewalls" at the MailChimp blog:

http://mailchimp.blogs.com

Challenge/Response Filters.

These are more common among "at-home" recipients (because they're too intrusive to use at work). When you send email to someone with a challenge/response filter, here's what happens. If you are not already in that person's "buddy list" or "address book," then you're considered a stranger to him. And if you're a stranger, you *could* be a spammer. So their challenge/response filter sends you an automatic reply with a question that you have to answer, or some link you have to click (this is to prove you're a human, and not a spambot). The screenshot you see to the right is an example of a typical challenge/response reply, from Spamarrest, a very popular vendor.
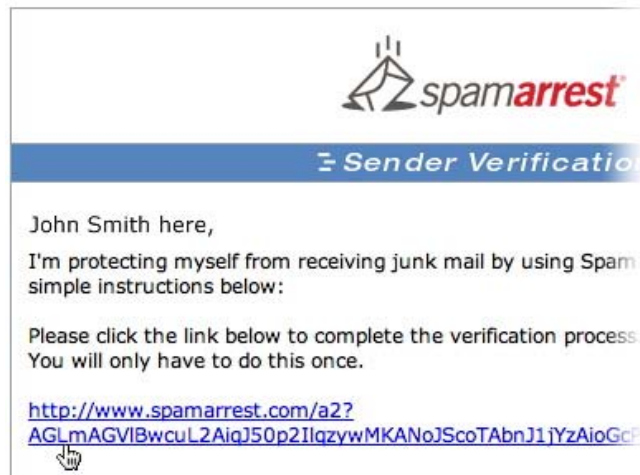
The thing to remember here is that you have to be "white listed" if you want your emails to get through. So when people fill out your opt-in forms on your website, ask them to "please add our email address to your address book..." Use your opt-in process as a way to "set expectations" and get "whitelisted" up front. Whenever you send a newsletter, make sure the "reply-to" address that you use is valid, and that a human checks it after each campaign. You can expect to receive a couple auto-replies like this after every campaign you send. Whenever we send our MailChimp Monkeywrench newsletter, we get about half a dozen of these autoreplies. I have to manually click each one of them if I want the email to get delivered.

Tactics for Avoiding Spam Filters

So now you know how a lot of anti-spam systems work. You really have to "think like a spam filter" when you design your emails. Just like you "think like a search engine" when you design web pages. Spam filters read your emails, and look for similarities with known spam. You don't want to do things that'll get you accidentally thrown into the junk folder.

And the best way to learn what **not** to do is:

1. Open up your junk email folder.
2. Look through all that spam you got. *Really* read it. Look at how they design things, and look at how they write their copy. Notice all the similarities?
3. Now, <u>don't</u> do what they do.

Okay, okay, to be more specific, here's some stuff to avoid:

- Using **bright red fonts,** or going crazy with colors, styles, and formatting. Check out the article, "<u>Let's dissect some spam!</u>" at our MonkeyBrains blog to see what I mean.
- Yelling with lots of exclamation points!!!!!!!
- YELLING WITH ALL CAPS!!!!
- Using spammy words, like "mortgage, viagra, etc."
- Saying, "free" or "click here!" or "click here now!" or "act now!" or "limited time!" (be especially careful of your unsubscibe link, where you might say something like, "click here to unsubscribe."
- If you use a WYSIWYG to code your HTML email, make sure the HTML code is "clean." WYSIWYGs often throw in lots of useless code that's invisible to the user, but spam filters look at it, and think you're a sloppy spammer who doesn't know HTML.
- Bad HTML in general can get you spam filtered. Missing table tags, content below the closing </HTML> tag, or empty <TITLE></TITLE> tags will get your message thrown into the junk folder. Don't get sloppy with that code! Note: the empty <TITLE> information is very common if you use WYSIWYGs. They throw that in by default, and expect that you go back and fill it in. Don't forget to go back and give the document a title!
- Don't use "dummy text" in your message, even if you're sending a test campaign. Spam filters get suspicious of all that "lorem ipsum" stuff. When sending test campaigns, use content that's as real as possible. Don't type in one

paragraph, then copy-paste it ten times. Spam filters can see all the duplicate content, and they think it's spam.

- Don't use the word "test" in your subject line.
- Don't get too creative with your spelling, because the spammers do that all the time (how many m0rtg4g3 ref1nanc3 spam messages are in your inbox right now?).
- Don't make an HTML email that's just a bunch of pretty graphics. You need some text in your message, too. Otherwise, the spam filters will have nothing to read, and will think your message is junk.
- Don't send the HTML email by itself. Always include that plain-text alternative message.

Learn More:

Read this article to learn more about how spam filters think:

http://www.mailchimp.com/resources/how_spam_filters_think.phtml

Testing & Troubleshooting Your Email Designs

After you've finally figured out how to design and code your first HTML email template (and, of course, your plain-text alternative version as well), don't just plop in your content and start sending campaigns to your list. You need to test your template first. Make sure it'll work in all the different email applications out there. Once you've found all the little bugs, and your template is "rock solid," you can start sending.

Web Designers: emails aren't the same as web pages!

If you're a web designer, you're probably used to testing web pages in a few different browsers, like Internet Explorer, Mozilla Firefox, and Mac's Safari. And you're probably familiar with a few annoying inconsistencies between all the browsers, and you have a couple hacks to make things look right.

Multiply all that by ten, for email design. There are tons of email applications out there that you need to test on, and they all render HTML email in their own annoying ways. Do you need to test every single newsletter across every single application, every single time you send? Of course not. Just setup a good, simple, rock solid template and test it as much as possible. Then all you do is swap out content. Here are some tips for testing your email designs...

Testing in Different Email Applications

In the past, the only way to test your campaigns was to setup a bunch of accounts at all the major ISPs, setup some test computers with different operating systems and email programs, and send test campaigns to them. It would take hours to do all this, and it really took all the fun out of email marketing.

That's why we created our Inbox Inspector tool. You build your campaign, and click one button. We'll generate screenshots of how your campaign will look in every major email program, and we'll tell you if any of the major spam filters and email firewalls will block your message, and why. We'll point out what content you need to fix in order to not get blocked. We'll even check for typos and HTML coding errors. Learn more about the Inbox Inspector at:

http://www.mailchimp.com/add-ons/inboxinspector/

If you prefer to test your campaigns yourself, here are some email applications you should install, and what to look for:

AOL

Their free CDs are *everywhere*, and they offer some very cheap $7 per month plans if you call and ask for them. They'll only give you 2 or 3 hours a month, but that's good enough for testing. AOL takes spam very seriously. So check to make sure your emails don't get filtered. Also, notice their "Report Spam" button. You've got to make sure your emails look relevant, or people will click that button (even if they opted in to your list!). At the time of this writing, AOL 9 has an extremely tiny "preview" pane. It's about 194 pixels wide, so make sure that your branding shows up okay under those circumstances. If your email is too wide, and your logo is right-aligned, your recipients will never see it, and they may never realize who sent it.
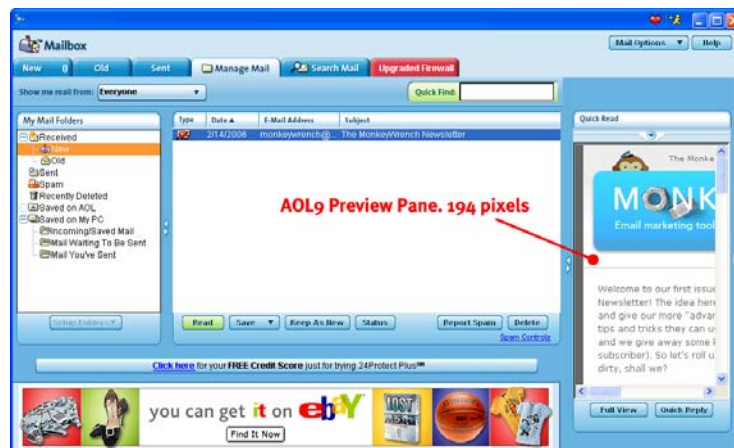
AOL's "preview pane" is *tiny!* Does your logo peek out enough?

Apple Mail

This is Apple's free email application. You get a nice big preview pane here, and they don't do much image blocking at all (yet). Images seem to be blocked only if an email ends up in its junk folder. Apple Mail comes with its own spam filtering system that "learns" so you should test your emails for any deliverability issues.

Apple's Entourage

Kind of like Microsoft Outlook, but for the Mac (it comes with Mac Office). Their HTML email support is fine, but we noticed in earlier versions of Entourage, if you send an

email in HTML format, but it consists of nothing but text and hyperlinks (no graphics or tables), then the links actually display their URL next to them. Kind of odd. But if the email actually contains a graphic or table, then the links work as coded. The issue seems to be fixed in the latest version of Entourage 2004.

Microsoft Outlook 2003

Outlook 2003 has better spam protection, so watch whether or not your emails get filtered. Also, this one's got 3 vertical panes, so your emails have less room in the preview area (see below).
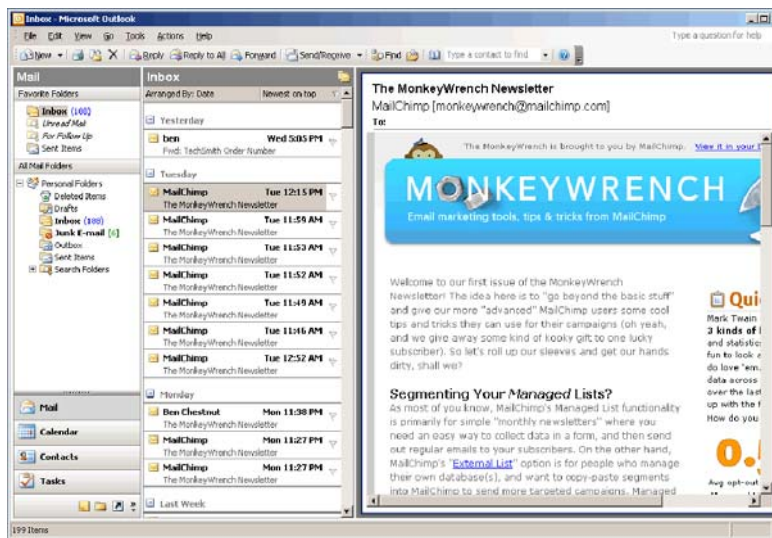
Three vertical panes in Outlook 2003 leave little room: Make sure you fit!

Microsoft Outlook 2000

There are lots of business users who haven't upgraded to Outlook 2003 yet. This version of Outlook predates all the vicious worms, spam, and viruses that we're used to now, so more of your emails with flash and videos and crazy stuff will actually work here. Don't let it build your confidence. That fancy stuff won't work in other programs.

Microsoft Outlook Express

This is the free version of Microsoft Outlook, installed on all Windows machines. Used mostly by "at home" recipients. Not really many HTML email issues, besides the usual "block images by default" feature that's on just about every email application these days.

Microsoft Outlook 2007

For Outlook 2007, the big change is that Microsoft is switching from using Internet Explorer (a web browser) to render HTML email to using Microsoft Word (a text editor) to render HTML email. Obviously, HTML email would render a lot better in a browser than Microsoft Word, so this is kind of a bummer. Microsoft says they're doing this because they want their customers to have more control over editing HTML email. Whatever. Here's what you need to know about Outlook 2007:

- Background images won't work.
- Most CSS won't work.
- Flash definitely won't work.
- No forms
- No animated Gifs

Lotus Notes 6.5.3 and 6.5.4

If lots of your recipients work at very large companies, you'll need to check your emails in Lotus Notes. This is probably the most frustrating email application out there. It does the strangest things. You'll need to keep your layouts very simple. Even still, Lotus will find a way to mess things up. I'm still baffled about how they managed to make a cell in my table blink red. There are even big differences between Lotus Notes 6.5.3, vs. 6.5.4. Go to our MonkeyBrains blog, and search for "Lotus Notes" for all the different things that can go wrong in Lotus.

Eudora 6.2

Eudora has good, predictable HTML email support. We don't see too many issues to talk about here.

Mozilla Thunderbird

A free email application that's gaining in popularity. Does some odd things with alt-text, and image blocking (search for "mozilla thunderbird" at the MailChimp Blog), but for the most part supports HTML email great. Thunderbird 1.5 has a feature that tries to detect "scams," where they look for hyperlinks that claim to point to a website, but the code actually links you somewhere else. If it detects a potential "scam," you get a warning across the top of the message, saying "Thunderbird thinks this message is a scam." You can click the "Not a scam" button if it was a false alarm. It's a nice idea, but it seems to give lots of false positives, especially when you use click tracking in your emails (all your links will point to a redirect script, in order to track clicks). In the Thunderbird support forums, they say that the feature is still kind of new, and they're working on refining it.

We don't think it's much of a problem, as long as your emails look nice and professional.

Testing Browser-based Email Services

You might want to setup accounts with all the browser-based email services. They're usually free, so it's easy to setup lots of test accounts. You *especially* want to look at how browser based email services alter your HTML and CSS.

Also, since these services are used by bazillions of people, their anti-spam filters are set really high. We listed them below in order of most-to-least popularity (according to a recent survey across all MailChimp managed lists). When you setup your test accounts with these services, leave their "junk mail filter" settings to "default." It really pays to test on these accounts before sending your campaign...

Yahoo!Mail

Pretty aggressive anti-spam filters (understandably). Other than that, they do the typical browser-based email stuff, like stripping your BODY and HEAD tags.

Yahoo!Mail Beta

A much richer browser interface, with a preview pane. So far, support for HTML email looks great.

Hotmail

Similar issues as Yahoo!Mail, where BODY and HEAD tags are stripped, so background colors and embedded CSS are lost. At one time, Hotmail was "expiring" hyperlinks in emails that were open for 5 minutes or longer.

Gmail

Extremely aggressive spam filters. And very, very little CSS support. It strips embedded CSS, (even if you keep the CSS out of the <HEAD> tag). It also strips every "class" code that it finds (such as in <span class="header">). To get CSS to work in Gmail, you'll need to use inline styles. Otherwise, your fonts will default to Arial, colored black, and about 13 pixels in size. If your fonts don't adhere to your CSS rules, check to see if they end up blowing out your tables cells, or wrapping in weird ways.

Check Different ISPs, too

If possible, check your emails when they're sent through **different ISPs**. Different email servers will alter your messages *before* they even get to the recipient's email application. For instance, some ISPs use email servers that will strip any content below a line in your email that starts with a period (I know, weird, huh?). We've been surprised at how differently email looks when checked in Outlook 2003, but received through:

- o Comcast
- o Bellsouth
- o Earthlink

Send tests to friends and co-workers

If you can't setup a few test computers, keep your designs really simple, and send each campaign to a few friends or office-mates. Just ask them to let you know if anything looks "off" or broken to them.

Or, just use MailChimp's Inbox Inspector

If you choose to setup your own test machine, and all those test accounts with the different ISPs, you could easily spend a couple grand. If you also want to check your campaigns against the big server firewalls (like IronPort, MessageLabs, and Postini) those are several thousand dollars just to get setup. If you've got the money, go for it. But who has the time to *maintain* all that stuff?

You can save yourself a ton of time and money by just using our Inbox Inspector. Here are some screenshots of what you'll get (it costs less than what you'd charge for one hour of your time):

We'll generate screenshots of your campaign in all the major email programs:

Zoom in to see where your email "scrolls" in each program, on a 1024x768 screen:

We'll tell you if spam filters are likely to block your campaign, and what you need to change if you want to get through:

Inbox Inspector is an optional add-on, which you can find under the "Account" tab in MailChimp. Learn more at: http://www.mailchimp.com/add-ons/inboxinspector/

Email Marketing Basics & Best Practices

If you're new to email marketing, we should go over some basic principles before you start sending campaigns. We'll go over best practices, legal issues, and how to measure your overall performance...

The Definition of Spam

If you get an unsolicited email from someone you don't know, is that spam? Not necessarily. If you get an email that was obviously sent to a whole list of people, is that spam? Not necessarily. So what's spam?

Spam is when you send an unsolicited email to a whole list of people.

So let's say you just bought a list of email addresses from some local business organization. These are great prospects for your business, right? You want to send them an email with a relevant offer they can't refuse…

It's spam if you upload that list into MailChimp (or any other email service provider like us) and send that list an unsolicited email.

It's *not* spam if you take that list, and write personal, one-to-one emails to each recipient, and the content is unique for each recipient.

If your immediate reaction is, "but what if…" you will probably get yourself reported for spamming. You simply cannot send unsolicited email to a list of people you don't know.

The CAN-SPAM Act of 2003

The United States federal CAN-SPAM Act became law on January 1, 2004. According to their website, the FTC says that if you violate the law, you could be fined $11,000 for *each* offense (multiply $11,000 times the number of people on your recipient list). ISPs around the country have already successfully sued spammers for millions and millions of dollars under this new law.

We can't give you much legal advice; MailChimp's an ape, not a lawyer (okay, insert lawyer joke here). But if you send commercial email, you should read through the CAN-SPAM Act of 2003, and understand the rules. If you have a lawyer, consult with her. Here are a couple of points we'd like to highlight:

- If you're sending "**commercial**" email (where you're selling or promoting stuff), here are just a few rules you should know about:
  - Never use deceptive headers, from-names, reply-tos, or subject lines.
  - You must always provide an unsubscribe link
  - Remove recipients from your list within 10 business days
  - That unsubscribe link must work for at least 30 days after sending your email
  - You must include your physical mailing address in the email (PO Boxes are not sufficient)
  - To learn more, go to: http://www.ftc.gov/bcp/conline/pubs/buspubs/canspam.htm

Royal Screw-ups To Avoid

Look, if you've downloaded this guide, and you've actually read this far, you're obviously interested in doing the right thing. You're probably not an evil spammer. But even the most well-intentioned legitimate email marketer gets reported for spamming. Do you know how you get reported? Some people think it requires a lot of work. Actually, all it takes is for one recipient to click a little button in his email program.

When he clicks his "this is junk" or "report spam" button, a little alert gets sent to his ISP. Then that ISP sends a warning to the sender (so long as the sender has responsibly signed up for that ISP's feedback loop). If an ISP sends you 1-5 warnings, you're in trouble. 5-10, and you can expect your emails to get throttled. More than that (some of them say less than 1% of your list) and your email server gets blocked. None of your

emails will get through to that ISP anymore (unless you can provide proof of opt-in for those complainers).

And ISPs aren't just blocking email servers anymore. They're scanning reported spam for URLs and domain names. If they find your company's domain name in reported spam, they'll block any future emails they receive with that domain name in it (no matter where it was sent from, or who sent it).

So it's very easy to get blacklisted. When that happens, it can take months to get your name cleared (if ever). Usually, companies get blacklisted because they made some simple little (avoidable) mistake.

We've been in this business for years, and we've seen thousands of marketers send millions of emails. Needless to say, we've seen lots of screw ups. Here are some of my most memorable examples. Learn from them.

Some of the details have been changed, to protect identities…

"But these are co-members of a local organization"
Our first "abuse desk situation" was from an alpaca farmer. He was a really nice guy. He was sending a beautiful email newsletter with alpaca breeding tips to members of an alpaca farming organization (yes, those exist). Problem was, those members never asked for his email newsletter. He got their email addresses from the head of the organization. Surely other Alpaca farmers would love hearing about this stuff, right? Not if one of his competitors is on the list. One of his recipients didn't appreciate getting emails from a competitor, so he repeatedly reported the newsletter as spam to his ISP (and to us). Since our customer was sending emails to a 3rd party list (strictly forbidden on MailChimp), we had to shut his account down. See above: *definition of spam*.

"But they dropped their business card in my fish bowl."
We had a real estate agency sending emails to a list of recipients about a brand new condo property that was opening up. As soon as they sent their campaign, we started to receive complaints. When we asked them where they got their list, turns out they setup a fish bowl at local eateries, for a chance to win a free lunch. No mention of any email newsletters, or opting-in to any lists, was placed on those fishbowls. We had to shut them down. This one's notable because their web developers threatened us, cursed at

us, and called *us* the spammers. They actually defended themselves by saying, "C'mon, they dropped their cards into a fishbowl. They knew what they were getting in to."

"But we swapped business cards 5 years ago" or "But these are in my CRM"
Unfortunately, this is a recurring one. Sales guy sets up a booth at a tradeshow, and swaps cards with someone. That someone is now a "prospect." That prospect gets entered into their CRM. It sits and waits. And waits. One year later, the company decides to send an email newsletter. The prospect is added to the list, and gets a huge, 5 page HTML email newsletter with all kinds of promotional junk that he never requested. The prospect clicks his "report spam" button. This is probably the 2nd most popular reason we shut down accounts (next to real estate agents with fishbowls). If you're a Marketing Director, and you've decided to add email marketing to your strategy, that's smart. But don't go to your sales team and ask them, "alright, gimme every prospect you've ever talked to, so I can blast our newsletter to them."

"But this list was really, really expensive"
A creative agency purchased a list of email addresses from a leading resource for advertising professionals (we won't name the resource). The people in that list were members who voluntarily submitted their contact information, just for networking purposes. Not for receiving emails from 3rd parties. By the way, this is almost always the case when you buy email lists---those people never gave their permission to have their info resold to you. Buyer beware. Anyway, this creative agency used MailChimp to send a "portfolio" email to that list. But one of the email addresses on that list was a "sales@" address, which means it got forwarded to every single sales employee at a pretty big, well-known company. A pretty big, well-known anti-spam company, to be exact. That customer was shut down immediately, and they got a very good lecture. Fortunately, the leader of the anti-spam group actually forgave that company, and we were able to reinstate their account. They learned their lesson: do not send mass, unsolicited emails to a whole list. If you're gonna buy a list, send your emails to each recipient separately, personally, and one at a time.

"But I'm sending for the government"
We had an organization who basically published scientific research papers online, into a searchable database. Where'd they get the research papers? Apparently, from the U.S. government. So where'd they get their email addresses? Apparently they were taking author names and guessing at what their email addresses would be. They'd send emails like, "Congratulations, Professor John Smith! We've published your research paper at XYZ

website!" Problem was, they sent to the wrong Professor John Smith. Professors from universities all over America were reporting them to us for spamming. Not sure how they did it, but we had to shut them down. Their reply? "This is not spam. This was sent for the U.S. Government." Sigh.

"Remember me from 10 years ago?"
One of our users sent a software "update" to his list of about 20,000 customers. These were people who had downloaded and purchased his software. 10 years ago. Man, was the Internet around back then? Anyways, to his credit, he had a very good, "Hey, remember me, from 10 years ago?" introduction paragraph and he even merged in each recipient's order information: Full name, how much they paid, order ID#, etc. He really went all out with the details. Unfortunately, 10 years is a long time to wait to send email updates. Almost none of those emails were even valid anymore. And some had no doubt been converted into spam traps (Microsoft does that with old, expired email addresses sometimes). As soon as he sent his campaign, one of MailChimp's IPs got blacklisted by Hotmail and MSN. We had to reroute outgoing emails from those IPs, and work with Microsoft to explain the situation. Luckily, the sender had overwhelming evidence in his campaign that they were his customers, and we got delisted.

The old Outlook Address Book dump.
A nice woman from a small town in North Carolina sent out an email campaign for her local beauty salon. It invited women to "come in and get a manicure." After she sent her campaign, I got an email from a very, very angry **man** about how "this woman is using MailChimp to spam me." I investigated the man's email domain, and noticed it was from an ISP located in the same small town as the sender. That's too much of a coincidence for me. I asked him, "any chance your wife signed up for this newsletter?" Nope. No wife. No kids. And nobody who would ever have access to his computer. "She obviously purchased an email list from somewhere," he tells me. Now, if some local business owner sends an email to 3,000,000 recipients, that idiot bought a list. But this woman sent to a couple dozen people, with no other complaints. Hmm. He also tells me that his email address has been dormant for years, and he was shocked to even be receiving any messages to it. After many hours of back and forth conversations with the sender and the complainer, we figured it out. She dumped her entire Outlook Address book into her email list. She figured the only people in her address book were her clients, since that was her business computer. So how did he end up in her address book? He used to be the tech support admin for her ISP, before they were bought out by a bigger ISP. You really shouldn't dump contacts from your address book into a newsletter list. Those

people might be your friends or clients, but they haven't necessarily opted-in to receive email newsletters from you. You can take the time

"But my sponsor asked me to."
One customer implemented a double opt-in list, and had a very nice, respectable looking email newsletter design. He even had a nice, well-branded welcome email. He sent a campaign out one day and got reported as a spammer by over 60 of his subscribers (usually, 1-5 reports is enough to get blacklisted). What would compel so many people, who specifically requested emails from this man, to then report him for spamming? Turns out for this particular campaign, he removed his logo, and changed the design dramatically. Instead of a white background with his orange and blue logo at the top, he sent an all-black design, with white and red text. Why? A company was sponsoring his email campaign, and that company wanted him to follow *their* branding guidelines, not *his*. Sigh. If you're sending to *your* list, make it look like you sent it (and tell your advertisers it's for their own good too).

Double Opt-in

We *highly* recommend the double opt-in method when managing your email lists. In fact, it's the *only* way MailChimp's built-in list management system will work. Here's a quick overview of the process:

1. A customer signs up for your email newsletter through a form at your website.
2. He receives an email with a confirmation link.
3. If they click the link, they're added to your list, and you store the IP address, date, and time of registration. Now you've got proof of opt-in, should you ever need it in the future (like if you receive a false or malicious abuse report).
4. If he doesn't click the link, he's not added to the list.

Double opt-in is fast replacing the (single) opt-in method, where someone submits a form, and bam---they're added to a list. There are too many chances for someone to get signed up to a list without his permission, either erroneously or maliciously. And there's no need to even discuss the old opt-out method anymore. That's getting phased out due to all the spam complaints marketers get, from people who never saw the opt-out check. Don't be so desperate to grow your list that you put your company's reputation on the line.

Pros & Cons

There was once a big concern about double opt-in, where marketers complained, "but nobody's going to click the confirmation link! I'll lose all my prospects!" This isn't a huge issue anymore. People are used to the double opt-in method these days. Plus, it just looks more professional.

Sure, if you compare a double opt-in list to a *single* opt-in list side-by-side, the single opt-in list will be larger. It's just too easy to subscribe (anyone) to a single opt-in list. But it will also have more "accidental" members, a higher bounce and unsubscribe rate, higher abuse complaint rate, and lower response rate. Double opt-in ensures that every single recipient truly *wants* to hear from you.

Measuring Performance

After you become a pro at HTML email design and coding, and you're sending campaigns on a regular basis, how do you measure performance? How do you know if your email marketing efforts are working or not? How can you experiment, and improve your performance? Here's a basic rundown of what you should be measuring, and how you can test your campaigns.

Open Rates

How many people opened your email? An open rate of about 20% to 30% is considered "average." Watch your open rate over time. Your first email might get a great open rate (like 50% to 70% or more), but over time will level off some.

Try experimenting with different things, like:

- Your subject line: Make it longer (or shorter). Merge their first name into the subject. Use your company's name.
- Date of send. Do weekdays work best? Or is the email geared towards an "at-home" audience, who check emails on weekends? Or, do they typically only order while at work, on fast connections?
- Time of send. Mornings, just before the first wave of spam? Just before lunch? Just after lunch? Just before quitting time? If you were a restaurant owner, would you send a campaign for a Friday event on Monday, or Thursday?

We analyzed over 40 million emails sent from our system, and found that subject lines that contain your company name in them, or that are seemingly very plain and "boring" result in better open rates than subject lines like, "FREE OFFER INSIDE! OPEN NOW!" Subject lines should tell what's inside, not sell what's inside. For details of our subject line study, look in the MailChimp Resource Center: http://www.mailchimp.com/resources/

Click Rates

How many people clicked links in your email? Which links did they click the most? Did they click on product links, or research links? Did you see a rise in purchases? How long after you sent the campaign do links keep getting clicked?

Try experimenting with:
- The number of links in your message. One big, giant link, or lots and lots of different links?
- Product photos for each link
- The wording of each link.
- Segmenting your list, and sending more focused emails (and links that are more relevant)

Example of a typical MailChimp report: You can see which links people are clicking...

Clicks by URL		
URL	Total Clicks	Uniques
http://www.acme-ind.com/index.html	67	55
http://www.acme-ind.com/landing/preview.html	28	25
http://www.acme-ind.com/updates/refer.phtml	19	17
http://www.acme-ind.com/updates/unsubscribe.html	15	13
http://www.acme-ind.com/privacy.html	2	1

© Copyright 2005 The Rocket Science Group, LLC. All Rights Reserved.

Unsubscribe Rate

What's your unsubscribe rate after each campaign? Less than one percent is average for lists that are contacted regularly, and well-maintained. If you send very infrequently, or if it's your very first send, your unsubscribe rate may be much higher. Check your unsub

List Summary - Acme Bananas

List Stats:		Since Last Sen	
Current Members:	147 (view all)	Signups:	0
Unsubscribes:	8 (view all)	Unsubscribes:	0
Cleaned:	0	Cleaned:	0

rate after each campaign. If you see it spike after a particular campaign, see if it had anything to do with your content. Maybe you're sending too frequently. Maybe not

Bounce Stats	
Total Bounces:	2 (1.47% of 136 sent)
Syntax Errors:	0
Soft Bounces:	2 (1.47% of 136 sent) Email Headers
Hard Bounces:	0

frequently enough. Keeping a good, fresh, quality list will result in a very low unsub rate.

Bouncebacks

Watch your bounceback rate after each campaign. A good list management system will break down your bouncebacks into "hard" vs. "soft" bounces, and clean your list for you. Soft bounces are emails that exist, but for some reason, they couldn't be delivered. For instance, their server might have been too busy at the time of your delivery. Hard bounces are undeliverable---perhaps the email account doesn't exist anymore, or there was a typo in the address. Hard bounces should be removed immediately. Your email marketing service should let you read the header information from each bounced message, so you can see exactly why they bounced.

Try experimenting with:
- Re-sending your campaign (a little later) to the people who "soft-bounced."
- Look through your hard bounces for any obvious typos. For instance, "hotmial.com" could probably be corrected to "hotmail.com"

MailChimp Helps Manage Your Bounces

Not all bounce-backs are bad. Soft bounces are mostly "temporary" and you can probably keep them on your list and try again. If MailChimp sends your campaign and an email address "soft bounces" back to our system, we'll watch that email address over the next 5 campaigns. If it keeps soft-bouncing, we'll automatically clean it from your list. But if an email address "hard bounces" (Like when it's an undeliverable address), we *immediately* remove it from your list. You shouldn't send campaigns to a bad email address over and over---ISPs block you for doing stuff like that.

Viewing SMTP replies for bounced email

When a recipient's email address bounces back to MailChimp, the server that bounced it usually sticks a little "message" in the email's header, which is called the "SMTP Reply." These SMTP replies tell us why the email was bounced (such as

Visit http://www.MailChimp.com

To learn the **exact** reason each email bounced, click to read their full

Soft Bounce Records

Bounced Recipient:
1 jim@acmebananas.com
2 john-doe@acmebananas.com

© Copyright 2005 The Rocket Science

Delivered-To: Autorespond
To: bounce-183914-john-doe
From: john-doe@acmebanana
Subject: Out of the offic

I am currently on an exte
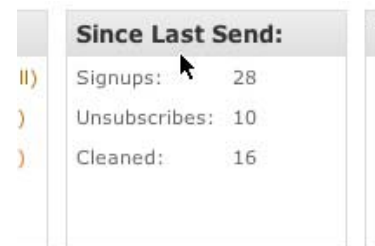-------- Original Message

50

temporarily busy, overquota, on vacation, bad email address, account no longer exists, etc). This helps MailChimp classify the bounce as "soft" or "hard." But sometimes, email servers don't use standard codes and messages in their SMTP replies. It's all up to how the server admin set things up. Worse yet, some anti-spam firewalls outright **lie**, by inserting "this email address no longer exists" into their SMTP replies. This is meant to trick list management systems into removing an email address from their list. Classifying bounce-backs as "soft" or "hard" isn't a perfect system. That's why MailChimp gives you the "View Full Headers" option in your bounce reports, so you can read the headers of each and every bounce-back, and determine exactly why each email bounced.

Traffic to your website

Check your website traffic logs after each email campaign. Does traffic pick up? Do orders increase? Do the cash registers go, "cha-ching?" Check and see if you get a spike in traffic immediately, or if it's more like a "gradual wave." How long does the new traffic last (and how long should you keep the graphics and pages that your email points to hosted live)?

Signups since last campaign

After each campaign, do you get lots of new subscribers? That could mean your wonderful recipients are forwarding your emails on to friends. How nice! Your content is actually useful! Don't see any list growth at all? Make that content more interesting!

Try experimenting with:
- Place some text near the top of your email, that says, "feel free to forward this to a friend"
- Place a "view this email in your browser" link at the top of your email. It should point to an archived version of the campaign on your website. Some people like to forward *that* link on to their friends.

MailChimp Campaign Reports

MailChimp can generate reports for each and every campaign that you send. We track open rates, URLs that were clicked, bouncebacks, unsubscribes, and more. You can also download every report into Microsoft Excel format, and generate your own graphs, or send them on to clients.

http://www.mailchimp.com/learnmore.phtml

Wrapping Up

So there you have it. Everything a beginner needs to know about designing, coding, sending, and managing email campaigns. We hope this guide helps get you on your way to sending many successful campaigns. Following is a "quick list" of the overall email newsletter creation process. Use it to send your very first email campaign (whether you use MailChimp or not). Finally, be sure to keep your skills honed by visiting all the useful email marketing websites and blogs that are out there now (we included a list of favorites in the Appendix).

"My First Email Campaign" Checklist for Beginners

- ❏ Decide how many lists you want. For example, you might want a "Newsletters" list, but also a separate "Press & Media Relations" list. Some people also setup "Weekly Specials" list, and "Internal Employee Newsletter" lists. The basic idea is that you don't want to send your weekly specials to an entire list of people, when someone on your list might have just been expecting an occasional newsletter. In MailChimp, you can setup as many lists as you want, and we'll manage them free.
- ❏ Decide on your content and frequency. Will you be sending email newsletters every single month? Do you truly have the time to do that? What will the newsletter be about? Why would anybody want to subscribe to it? What will recipients gain from it?
- ❏ Decide on the "personality" or "brand" of your email campaigns. Are you sending lots of e-commerce specials? Or letters from the president? Personal? Corporate?
- ❏ Get permission. Ask the customers in your database for permission to send them email newsletters, offers, and promotions. Don't ever start "blasting" newsletters and email offers without permission.
- ❏ Make sure your privacy policy is in order. In particular, issues like "tracking personally identifiable information" or "marketing to children under 13." If you're tracking opens and clicks, and if it's possible for children to subscribe to your emails, you need to cover those topics in your privacy policy.
- ❏ Know the laws. Become familiar with the CAN-SPAM rules. Consult with your attorney.
- ❏ Setup email addresses that you can use as the "Reply-to:" for your campaigns. For our MailChimp company newsletter, we use newsletter@. For our MonkeyWrench newsletter, we use monkeywrench@. You might use sales@ or

email@. Whatever. Just be sure it works, and be sure a human checks that account. If you use jenny@yourcompany.com, what happens when Jenny leaves the company? You'll need to change names to the new guy, but spam filters will have already been trained to receive emails from Jenny. Setup that sales@ email address from the beginning.

❑ Setup an abuse@ email account. No matter how clean you keep your list, you will always get complaints. It is *inevitable.* People forget they opted in. People get tired of hearing from you. People get mad from some lost shipment, or bad customer service. When that happens, they sometimes report your email as spam. If they subscribe to anti-spam services, or if they report it to their ISP, those representatives will want to contact you via email. And the 1st place they'll try is abuse@your-company-domain.com. So make sure that abuse@ works. And while you're at it, you should register that abuse@ address online (go to http://www.abuse.net/), because if/when they receive complaints about your emails, anti-spam organizations often use this to gauge how responsible your company is.

❑ Setup "feedback loops" with major ISPs. If you're going to send lots of email marketing **from your own company server**, you should register your IP addresses with AOL, and MSN so that they can send you a report if/when their users report your emails as spam. Google the term, "AOL feedback loop" for more information on how this works. MSN and other ISPs are said to be implementing similar reporting mechanisms. You should also go ahead and register for reports at http://www.spamcop.net. If users on their system report your emails as spam, they'll send you automatic email alerts. **If you use MailChimp** to send your campaigns. Don't worry---we've already got this covered. All our IP addresses for all our servers are registered, so we can handle any deliverability problems as they arise. If you use another service besides MailChimp, make sure they monitor this information too. Most reputable vendors do this.

❑ Setup test email accounts with Yahoo, Gmail, Hotmail, and AOL. Setup a test computer (or two) in the office, and download as many different email applications you can find, like Outlook, Mozilla Thunderbird, Lotus, Apple Mail, Entourage, and Eudora. You'll be doing tons of testing. Or, like we've said, try our built-in Inbox Inspector.

❑ Setup a double opt-in form on your website. Tip: create questions you might be able to use for future segmenting, like "industry" or "title" or "referral." Be sure to customize your confirmation screens and confirmation emails appropriately. Use

them to [set expectations](#) about the emails they'll be receiving. Want a working example? [Subscribe to our MailChimp MonkeyWrench newsletter](#) just to see how we do it (then unsubscribe immediately, if you want). After you get it set up, you can start inviting your customers to opt-in. Post links all over your website, your invoices, and in your email signature.

❑ Design your HTML Email template. If you're sending different kinds of communications (sales promotions, vs. monthly newsletters, vs. how-to's and tips), you may want to setup multiple templates for each occasion. If you're not the "design-it-yourself" type, you might want to try our built-in template designer. You can use it (free) to setup as many different email templates as you want, for any occasion.

❑ Insert real content into your test campaigns. Don't just use "lorem ipsum dummy text" when you test your campaigns. It's not accurate, plus spam filters often throw away emails with garbled, nonsensical text in them.

❑ Build your plain-text alternative email. Plain-text email has its own peculiarities, so you'll want to get a "template" for it refined from the beginning. Don't let plain-text be an afterthought.

❑ Test your templates. And we mean **really test them** like you've never tested before. Send them to friends, family, and colleagues. Try the templates in as many different email applications you can. Open your own emails, and click them like crazy. Now go check your reports. Make sure all the tracking works like it should. Hit the "reply" button and see if you get replies to your account. You want to uncover any embarrassing mistakes before sending to your *real* customers.

❑ Setup a test list. If you're like most people, you probably have about 3 or 4 email addresses. Why not setup a test list in MailChimp, and send your full campaign to that list? That's the best way to simulate what your email will actually look like when you send to your real list.

❑ On the subject of lists, make sure that **YOU** are subscribed to your own list. Even though systems like MailChimp allow you to send tests, it's best to be on the actual list, just like your customers.

❑ Prepare your website and landing pages for delivery. Before you send, you need to get everything in order. Did you create a copy/archive of the email for your website? Make sure that's all in place. Are all the landing or e-commerce pages that your email links to live, and working?

❑ Prepare the office. **What time** will you be sending your email? Here's a funny and embarrassing story. Once, we spent hours and hours prepping an email

campaign for MailChimp. We sent it out at 9am EST, and got tons of calls and orders immediately after. It was great! So our entire office went to lunch together that day, to celebrate. But "lunchtime" on the East Coast is "just getting into work time" on the West Coast! We all came back to tons of angry voicemails and emails from our West Coast customers waiting for us. D'oh! Now, we've got these cool clocks across our wall, representing every major time zone around the globe. Learn from our mistake, and be sure you'll be near the phone/email after you send. Let your staff know about the campaign too, if they take orders via phone, email, or live chat.

❑ Deliver the campaign (finally!).

❑ After the campaign, it's always fun to log in to MailChimp, because we provide stats in real time. You can virtually watch the opens and clicks as they happen! It's always shocking to see how many people open and click in the first few minutes after you send. Then, it trails off. You'll still get opens and clicks weeks after! That's why MailChimp keeps your links and tracking alive for three months before archiving them. And in our reports, we provide "Date of most recent open/click."

❑ Review performance. After a few days, go back and analyze your reports. How many opens? How many clicks? Compare it to your web traffic logs, and most importantly, to sales. Did the email generate leads or sales? What was the ROI on that campaign?

❑ Plan the next campaign. Use your campaign reports to understand what people clicked, and what they didn't. That should serve as the foundation for your next campaign. Maybe you'll split up your list and segment next time around, or test different subject lines or delivery schedules. Get scientific!

Appendix

Useful Resources

Here are some links to handy resources on the Internet for you

Email Marketing Sites

- http://www.emailsherpa.com - Great email marketing case studies and research. Be sure to sign up for their email newsletters.
- http://www.clickz.com - Lots and lots of online marketing and email marketing news.
- http://www.marketingprofs.com - Nice case studies and articles here. Premium membership is actually worth it, in my opinion. They also issue a Buyers Guide to email vendors.
- http://www.btobonline.com - Email and direct mail news. Check their "email marketer insight" section.
- http://www.futurenowinc.com/ - Look for their little alien mascot, and sign up for their Grokdotcom newsletter. These guys specialize in conversion, and they know their stuff.
- http://www.marketingexperiments.com - Great experiments and case studies here.

Email & Marketing Related Blogs

- http://www.ducttapemarketing.com/weblog.php - John Jantsch's Duct Tape Marketing Blog is full of quick, simple, affordable marketing ideas for business. His book is a great read for new businesses who are just getting started.
- http://sethgodin.typepad.com/ - That familiar bald head chock full of handy marketing idea
- http://www.tompeters.com/ - As Tom would say, "Wow! Nice blog."
- http://emailmarketing.typepad.com/ - Tamara Gielen's blog on email marketing. She posts links to news all over the email marketing world.
- http://mailchimp.blogs.com - Gotta plug our own blog, "MonkeyBrains" has email design tips and tricks, hacks, troubleshooting tips, news, and more.
- http://www.email-marketing-reports.com/ - Mark Brownlow has posted hundreds of useful email marketing articles and links here.

Products & Tools

Text Editors for coding

- http://www.bbedit.com - BBEdit is a great text editor for the Mac. Use it to code HTML and plain-text emails. Or check out their TextWrangler product, a lite, free version of BBedit.
- http://www.notetab.com - NoteTab Pro is a great text editor for PC.
- http://www.macromedia.com/dreamweaver/ - Used by web design pros to build websites. You can use it to build HTML emails, too. Use the coding interface for HTML, but be careful of the WYSIWYG tool. It adds CSS and other things to your file (which you'll need to take out for HTML email)

FTP clients, for sending files to your server

- http://fetchsoftworks.com/ - FTP software for the Mac. ($25)
- http://www.cuteftp.com - Great FTP software for PC ($90, and worth every penny)

Graphic design programs

- http://www.macromedia.com/software/fireworks/ - Great for designing anything for the Web. I've got an old installation of Fireworks 4 that I still use religiously.
- http://www.adobe.com/products/photoshop/ - A must-have for serious graphic designers. A bit overkill for HTML Email design, but you can't list "graphic design programs" and not include Photoshop. Also check out Illustrator (http://www.adobe.com/products/illustrator/)

Miscellaneous

- http://www.apple.com/downloads/dashboard/reference/picker.html - A great Apple Dashboard widget for quickly picking websafe colors to use in your emails.
- http://www.vmware.com/ - We use vmware to setup testing computers, with multiple operating systems and email applications.
- http://www.newsletterarchive.org/ - Online repository of email newsletters. Great if you're looking for ideas and inspiration. Take a look at what some of the largest companies are doing with their email marketing
- http://www.templatekit.com - They sell tons and tons of templates for websites and emails

- http://www.mailchimp.com/resources/templates/ - We posted some free HTML email templates, which are easily tweaked to match your brand, and pre-tested on all the major email applications.
- http://www.mailchimp.com/help/ - Our Help Center, with searchable knowledge base.
- http://www.istockphoto.com - Really, really, really cheap stock photography. We get all our photos from here. It's a great site.

Email Applications & Known Issues

Following are the major email applications that we test new templates in, plus the known issues that we've uncovered for each. Use it as a starting point, but be sure to test thoroughly on your own. Many years ago, when we first published our HTML email design guide, we listed all kinds of issues across different email apps. But these days, they've all gotten a lot better (as long as you keep your layouts in simple tables, and don't use CSS-positioning, DIVs, or anything that sets off antivirus apps, like Flash, JavaScript, and ActiveX). This table has become more a list of "peculiarities" than concerns.

Email Application	Known Issues
AOL 9	**Nothing horrible here, but their "preview pane" is extremely tiny. Make sure to design your emails to peek out enough from the preview pane. Your HTML code needs an extra line break or two after the final </HTML> tag, or you'll see weird "equals signs" in your email**
Outlook 2000	This one's getting replaced by Outlook 2003, so won't be around much longer. Default installations of Outlook 2000 will display just about anything you throw at it, including embedded movies (hence all their virus and spyware problems). Displays HTML email just fine, save for some obscure CSS properties. Like AOL, if you don't have an extra line break under the closing </HTML> tag, you may see mysterious equals signs appearing in your email.
Outlook 2003	Blocks images by default (as do most apps

	these days) to protect your privacy. Displays HTML email fine, but default installations of Outlook 2003 open up in 3 vertical panes, leaving little room for your preview pane.
Mozilla Thunderbird	Thunderbird blocks images like other apps, but doesn't use big, obvious placeholders. If you use "alt text" for a logo graphic for instance, the Alt test will display, instead of a big "broken/blocked image" graphic. The downside of this is that, depending on what alt text you use, people may never know there's an image to display, and won't click the "show images" button. If you don't use alt text for your images, a placeholder seems to appear like other apps. Thunderbird 1.5 has a new "This looks like a Phishing scam" feature, where you get a big warning icon at the top of the email if it detects a link that appears to be deceptive. For instance, if it finds a link described as, "Paypal home page" but the URL actually points to some suspicious URL. Nice feature, but it gets triggered if your email uses click-tracking. Mozilla says they're working on improving it, but it's not a huge issue for you if your emails always look reputable.
Lotus Notes	Ah, Lotus Notes. Where do we begin? Actually, Lotus has gotten a lot better with HTML email, but they've still got peculiar issues that pop up now and then. You'll almost always get a random border box around something. A graphic, a table cell---

	whatever. Don't ask me why. It just happens. Oh, 6.5.3 handles center alignment ok, but 6.5.4 doesn't. Go figure, And neither seems to handle animated GIFs with transparency. COLSPANs can also be touchy here. Keep the table structures very simple.
Comcast, Bellsouth, and other "at-home" ISPs	We've noticed some "at-home" ISPs use email servers that interpret lines beginning with "periods" to mean "end of email." So be careful not to start any lines with a period. Sounds weird to do such a thing anyway, but places where it often occurs include: plain-text emails, where someone uses periods as visual markers and divider lines. Embedded CSS code (.header, .footer, etc). Indent your CSS, or add a space in front of each line to get around this. Just recently (February 2007) Bellsouth appeared to have made a change to their filtering, where they were blocking all HTML emails. After some testing, it seems they are only blocking HTML emails that are not authenticated. Our HTML email campaigns that were SenderID and DKIM authenticated made it through Bellsouth's servers in tact. If a large portion of your list uses Bellsouth, you might consider implementing authentication.
Outlook 2007	There was a big uproar after Microsoft announced they'd be changing Outlook 2007 to use Microsoft Word instead of Microsoft Internet Explorer to render HTML emails. You'd think that IE would be the

	smarter choice, since it's a web browser, and can theoretically display HTML emails better. But Microsoft defends their choice by pointing out their users need ways to edit HTML email, not just read it. Also, many of the limitations of Microsoft Word were limitations that already existed in other email programs. So we haven't really lost much functionality here.
Yahoo Mail Beta	Yahoo has upgraded their email service with a richer interface (they're using that fancy AJAX stuff). Their support for HTML email seems fine, except for the usual CSS quirkiness that comes from most webmail clients. We've heard that Yahoo plans to (finally) implement a feedback loop, which is good for people who want to keep their lists clean. But in order to qualify for the feedback loop, they say you'll need DKIM authentication installed. Unfortunately, DKIM is not a very easy thing to install. We'll see how this plays out.
Internet Explorer	If your recipients check email in their browser (like using Hotmail), chances are they're doing it in IE. We've noticed IE7 will break tables in your email if you forget to include a # symbol in your hex values. So if you specify something to be red with #FF0000 but accidentally enter FF0000 instead, your entire email could go out of whack.