

Before you begin reading ... PRINT the report out on paper. I assure you that you'll receive much more benefit from studying over the information, rather than simply browsing through it on your computer screen. So, I encourage you to print the report out before you read it.

#####

Coding HTML Email: Tips, Tricks and Best Practices



An HTML email is still the most successful means of communication between email marketers and subscribers. It allows marketers measure the recipients' interest in products or articles by tracking opens and clicks. For recipients, a HTML email is more visually appealing and is much easier to scan and navigate than a plain text message. However, together with all its advantages a HTML email is a big concern for email designers and programmers...

By Julia Gulevich,
www.glocksoft.com

Introduction

There are plenty of different email applications – from desktop programs to web-based services – capable to read HTML email and each of these applications can render the same HTML message in a totally different way. I hope the strategies described in this report will help you create an HTML email template that will work across a variety of email software.

Before we jump into details, I just want you to remember two basic concepts when you start coding a HTML email:

1. Use HTML tables to keep the email layout. You are probably using CSS layouts to code your web pages, but they will just not work in an email.
2. Use inline CSS for other components within your email, for example, background fonts and colors.

Using HTML Tables to Control Layout

First of all, you need to decide on your layout. For email newsletters, a one-column and a two-column layout are best. They allow place the content destined for the readers' attention in the best way.

A one-column layout generally includes:

- a header with a company logo, message title and navigation links from the main web site;
- links to articles or stories going further in the email;
- a footer which often has the same links that are in the header, contact information, privacy policy and unsubscribe/update profile instructions.

A two-column email also has a header and footer. One narrow column usually features navigation links. The other wider column contains the text of the email.

You can easily create these layouts and separate space into rows and columns using HTML tables. Basically, HTML tables are the only solution to create a layout that will display consistently in different email applications.

Remember that no matter which layout you choose, you should place the most important piece of content at the top or not far off the top of the email, so the reader sees it immediately after opening your message. The top left corner of the message is usually the first place people take a look at when they open a message.

After you decided on your email layout, use this approach to create an HTML email.

If you want a two-column layout, create three tables: one table for the header, one two-column table for the content and one table for the footer. Enclose these tables into one container table.

Do the same to create a single-column layout, but the table for the content should have a single column only.

A layout using a separate table with *td* rows for the header (with *colspan="2"* for a two-column layout), content, and footer should be rendered fine in email programs (except Lotus Notes).

Use the attributes within the table and *td* tags to define the table's parameters: *border="0"*, *valign="top"*, *align="left"* (or center, if that suits the design), *cellpadding="0"*, *cellspacing="0"*, etc.

While coding the email you may consider setting *border="1"* to debug any issues that may occur with the internal alignment of *tr* and *td* tags.

Then you can set it back to *border="0"* for email testing if the design of your email doesn't assume a border around your table.

Though this method may not be suitable for those who prefer coding HTML using the latest standards, it is the only technique that works at this point. Note that using tables to create the email layout doesn't mean you need to return to old methods completely. For example, no

matter how clumsy an HTML email displays in Lotus Notes, you should not resort to using the *font* tag. And MS Outlook 2007 which HTML email rendering engine is far from perfect displays simple HTML tables well.

Adding CSS Styles

Though CSS support is poor in email clients, you can still use CSS for styles in your HTML email after your layout is ready. Here are a few things to focus on.

Use inline styles to store all your style information, as shown below:

```
<p style="color: red;">
```

This includes *table*, *td*, *p*, *a*, and so on.

Do not put the CSS style declaration into the HTML *head* tag. But place your style declaration below the *body* tag. And don't use the link element to reference an external style sheet: Hotmail, Google Mail, and other email software will ignore, modify, or delete these external references.

For your container table set the width to 98%. Yahoo! mail seems to need that 1% gap on either side in order to display the message properly. You can even set the width to 95% or 90% to avoid potential problems. The width of the internal tables should be set to 100%.

Place general font style information in the table *td*, closest to the content. Place font style definitions into heading (e.g. *h1*, *h2*), *p*, or *a* tags only when necessary.

Use *divs* to float small boxes of content and links to the right or left inside a table's *td* cell. In particular, Google Mail seems to ignore the CSS float declaration while Yahoo! and Hotmail deal with it fine. Sometimes you should better code a more complex table layout than

rely on the float declaration. Or, ask your email designer to place the content that floats in the narrow side column instead. Unreliable support for floats can be a reason for an email design to be modified.

If *divs* seem to be rarely useful, *spans* work in all cases because they are inline elements. *Spans* can be used not only for colors and size. Sometimes, you can use them to put the text above or below the content.

Note that some email delivery services will unpack style definitions to make them more explicit and, therefore, more readable by all email software. It's recommended that you test each email in different email applications to see what happens to the HTML code.

Coding HTML Email for Outlook 2007 and Google Mail

MS Outlook 2007 and Google Mail present their own challenges for email designers.

Outlook 2007 has taken HTML email support backwards. Its HTML rendering engine is inferior in terms of CSS support in comparison with the older versions of Outlook.

Google Mail has its own particularities of rendering HTML. For example, Google Mail removes the CSS styles included between any style tags no matter where they are placed in the email. And fonts within HTML tables (the only alternative to styles) show larger than intended, no matter how the HTML email is structured.

Good news is that if you create an HTML email that displays well in Outlook 2007 and Google Mail, your email code will be more likely rendered well in many, if not all, email applications.

So, below are some approaches that seem to work in Google Mail and other older email clients:

- Set the background color in a *td* cell with the *bgcolor* attribute. Do not use the CSS style.
- For background images, use the *background* attribute in the *td* cell instead of using CSS. Using this technique you can make a background image as tall as you need. If your email template is going to vary in size, an extra-tall background image in this case allows the height of the email shorten or expand from one email copy to another. Note: Outlook 2007 ignores background images completely.
- Use the padding declaration to set margins within a *td* cell. While the margin style does not work in the cells, padding does.
- If you want to have a border around a *td* cell, remember that Google Mail shows a border around a *td* cell when the border is defined in a *div*. When the border is defined as a border style in a *td* tag, it won't be displayed in Google Mail.
- If you want to put a light-colored link on a dark background color, place the font definition in the *td* cell (so it applies to *p* and *a* tags equally) and then add a **color:** style to the *a* tag.
- If the *p* and *a* fonts show in different sizes, enclose the *a* tag in a *p* tag.
- Google Mail displays the email content in the middle panel of the user interface. So, set the padding style in the content *tds* to 10 pixels all round, so that the text does not slides apart the left and right edges.

- If you test HTML email with Google Mail, you are likely to find that one or more font styles are missing in the *td*, *h1*, *h2*, *p*, *a*, and other tags. Verify every font thoroughly to ensure Google Mail displays the fonts correctly.

Coding HTML Email for Lotus Notes

Besides Outlook 2007 and Google Mail, Lotus Notes is another application that presents a hazard for programmers concerning an HTML email design. For example, Lotus Notes can convert images to their initial formats, or simply ignore basic HTML code in one email, but render other HTML fine in another message.

The best approach to have an HTML email displayed well in Lotus Notes is to follow this rule: the simpler the HTML code, better it will work in Lotus Notes.

Below are a couple of tips that will help you make Lotus Notes to render the HTML code of your email properly:

- As I wrote before, use a container table to hold all the internal tables of your email layout – tables for the header, content, and footer. This allows keep the integrity of an HTML email, so the fragments of the layout don't crawl away when the email displayed in Lotus Notes.
- Use a cellpadding of at least 5 or set the width to a percentage to create a gutter around the container table.
- As mentioned earlier, do not use a *style* declaration in the *head* tag. This approach may work well for web pages but Lotus Notes as well as Google Mail may delete your styles. Instead, use inline styles within the *table*, *td*, *h1*, *h2*, *p*, *a*, and other tags.

- Specify absolute URLs to images that are stored on a web server, for example *http://www.yourdomain.com/images/pic1.gif*.
- Internal links in the email are not likely to work in Lotus Notes. Better if you avoid links that lead to a specific content further in the email.
- Avoid *colspans* in the HTML tables. Lotus Notes (especially its older versions) can cope only with basic table layouts.
- Make sure that the widths of the *td* cells are accurate because Lotus Notes sizes each *td* cell accordingly to its defined width.
- Centering an email layout usually doesn't work in Lotus Notes. Your email layout generally has to be left-aligned.

Using Best Email Practices

In addition, the following common best practices are recommended:

- Avoid using JavaScript. It works well for web pages but most email clients will disable it anyway.
- If an image is sliced up and spread across several HTML table cells, test the email using many test accounts. It may display well in Outlook but be shifted by one or more pixels in Hotmail and other services. Also consider making the image a background image on a new HTML table that encases all of the table rows and columns that would display parts of your background image. This often works as slicing an image up but uses less code and can provide better results. Note: Outlook 2007 does not display background images.

- For background images, use the table's *background* attribute instead of using CSS. This works more consistently across email clients than other solutions.
- Store the images on a web server, for example, in a folder called *http://www.yourdomain.com/images/* and don't delete them because some people open emails weeks or months later.
- Be sure all your images have the *alt*, *height*, and *width* attributes. Setting values for these attributes helps achieve better results in Google Mail, as well as maintain the email layout when a reader has their images turned off. Note: Outlook 2007 does not recognize the *alt* attribute.
- Use the *target="_blank"* attribute for the *a* tags, so that people who read the email with webmail services don't have the requested page appear within their webmail interface.
- Avoid using a large image "above the fold" in the email. This is a classic spammer practice and may cause your email to be considered as spam.

Testing HTML Email in Different Email Applications

Creating an HTML email using the guidelines I've suggested is only a half of the work. The next step is to test your HTML email and make sure it's displayed well enough in a variety of email clients. Testing often helps identify problems that require workarounds.

First, you should test your email in the web browsers such as Firefox and Internet Explorer. If the email displays well in both browsers, it

will likely be rendered well in Yahoo!, Google Mail, and other services (maybe you will encounter only minor problems).

Also, you'll do a good thing if you test your HTML email in Internet Explorer 6. You'll have a clear idea of how your email will render in Outlook 2003.

Once your email displays well in web browsers, test it in different email accounts. Ideally, you should setup test accounts with AOL, Yahoo!, Hotmail, and Google Mail. Then send your email to see how your HTML template is rendered by these services.

Here are the most common tweaks you may need to implement during the test phase:

- You may need to switch from percentage widths to fixed widths. Sometimes, using a fixed width is the only way to have a layout to display properly in multiple email clients. However, this is not ideal because readers can resize their email windows while reading the email.
- If there's a spacing issue with the columns in the email design, first tweak the *cellpadding* and *cellspacing* attributes of the HTML tables. If that doesn't work, apply CSS *margin* and *padding* attributes. HTML spacing works better with older email software.
- Images can be displaced if a *td* cell is closed below an *img* tag. Put the `</td>` tag right after (on the same line as) the *img* tag to eliminate the annoying 1-pixel gap.

It's important to ensure that your HTML email displays fairly well with images turned off. Thunderbird, Outlook and many other email applications are set not to display images by default.

To test how your email looks like with images off, you can replace each image's *src* attribute with a unique combination of characters, for example, *asrcb*, and then revert it back after the test.

After you tweaked your HTML email so that it displays well in most email clients and web services, the next step is to go through the final checklist and verify:

- Is the From field displayed as a name, not an email address? (At least use your company's email address that readers can easily recognize).
- Is the Subject line correct?
- Are there unsubscribe/update profile instructions?
- Is your contact information correct?

After you're gone through the checklist and verified all the aspects of your HTML email, you can hit "Send" in your email marketing software to send the email to the whole list.

Conclusion

Many people prefer receiving HTML emails over plain text messages. But for email designers, creating an HTML email that will display consistently in different email applications is a simple and complex task at the same time.

The main idea to take away from this report is - keep the HTML code as simple as possible. If you're hesitating between a simple and more complex email design, simplicity is always the right choice.

#####

Afterword

If you're not very good at HTML coding or don't have time to design a HTML email yourself, you can find a rich library of professionally designed HTML email templates at www.hotemailtemplates.com.

All the templates are tested with various email clients and browsers.

In addition, you can order a custom email template design if you don't find any template suitable for your needs. Our email designer will create a template based on your requirements within a few business days.

Free bonus from G-Lock Software:

350 Business Templates Collection

If you have a license for **G-Lock EasyMail v6 Business edition**, you can get 350 professionally designed email templates from www.hotemailtemplates.com all in one package at ZERO cost. The templates cost \$990 if you purchase each collection separately. Just send me your order ID for G-Lock EasyMail v6 Business edition at support@glocksoft.com and I will contact you with the instructions how you can get your bonus immediately.

If you don't have a license for G-Lock EasyMail v6 Business edition yet:

1. [Download a free version of G-Lock EasyMail](#)
2. [Upgrade to G-Lock EasyMail Business edition and get 350 business templates for free](#)