



An HTML Email Guide



***Definitive Results** helps marketing teams transform their operations through the power of marketing automation.*

DR Who?

Definitive Results is a boutique marketing consulting firm based out of Leesburg, Virginia that specializes in helping Marketers accelerate their efforts using the technologies available to them and alleviate their Marketing Automation pains aka “Eloquirks” or “Marketoerks” as we call them. We specialize in Marketing Automation, Marketing Operations, Marketing Analytics, and IT Services. Our mission is simple: we help Marketers transform their operations through the power of Marketing Automation. We love working with the next generation of modern marketers — those driving results based on data and analytic insights, not their guts or the wind.

- Our Experience: we’ve been in similar shoes to you. All of our consultants were previously users of a Marketing Automation tool(s) prior to joining DR giving us 20+ years of hands-on experience. So we really get your pains and challenges.
- Our Commitment: we commit to always respond to you within 1 business day throughout our relationship. No waiting and wondering.
- Our Rates and Flexibility: we offer highly competitive rates which we are transparent about from day 1 as shown on our How We Help You pages. If you don’t see what you need, we will create a custom plan and/or project for you.

We can plug into your team and become a key team member for you to count on whether its for strategic marketing projects or a simple task to just get done quickly in your Marketing Automation tool.

Contact us:

7 Loudoun Street Southeast, Suite #6, Leesburg, VA 20175

Phone: 1-855-MKTGDOC (658-4362)

E-Mail: info@definitive-results.com

Web: <http://definitive-results.com/>

Eloqua Email 101

Definitive Results

03.20.2014

Purpose

To identify common formatting issues when designing html emails for Eloqua.

Audience

Eloqua Users
Email Designers
Email Marketers

Sources

Email on Acid
C&A2011 Email Guide
EE12 - Do It - McAfee -
HTML Email Design

What is this and why should I read it?

This is a guide to provide information on the most common html email formatting issues for Eloqua and html email in general.

This resource is a compilation of information from various sources with the intent of creating a go-to-guide for common issues experienced in most browsers while creating html email for Eloqua. This guide will outline those issues and provide solutions to common problems to avoid frustration.

As to not reinvent the wheel here, I have found several excellent resources, which I will be citing.

Included in this guide will be quirks that we have experienced as well with designing, implementing, and modifying html emails in Eloqua. Some of this information you may already know, but we present it with the intent of being a thorough go-to guide for our internal use and training as well as part of our client-side sanity retention initiative.

Lets get started... HTML Emails are generated utilizing the HTML programming mark - up language. Different browsers and email programs handle and display HTML differently, resulting in inconsistent deliveries. This guide outlines best practices to ensure the best results in terms of deliverability, content and design.



Rule #1: Outlook is not your friend.

Designing html email for Outlook could lead to hours of staring at your screen, test after test, wondering "Why Outlook, Why?"

Luckily there are some common fixes for Outlook to ease this suffering. We will share these with you as well.

Just remember, Outlook is not your friend, it was apparently coded by Murphy's Law standards.



A great way to start your Email is with Email on Acids
 “Boilerplate” common fixes. Sure, we could just put a link here and
 a send you on your way. What kind of friend does that though?

(That being said it’s an awesome resource so you should check it out: <http://www.emailology.org/> - 1)

```
<head>
  <style type="text/css">

    /***** EMAIL CLIENT BUG FIXES - BEST NOT TO CHANGE THESE *****/

    .ExternalClass {width:100%;}

    /* Forces Outlook.com to display emails at full width */

    .ExternalClass, .ExternalClass p, .ExternalClass span, .ExternalClass font, .ExternalClass td,
    .ExternalClass div {
      line-height: 100%;}

    /* Forces Outlook.com to display normal line spacing, here is more on that:
    http://www.emailonacid.com/forum/viewthread/43/ */

    body {-webkit-text-size-adjust:none; -ms-text-size-adjust:none;}

    /* Prevents Webkit and Windows Mobile platforms from changing default font sizes. */

    body {margin:0; padding:0;}

    /* Resets all body margins and padding to 0 for good measure */

    table td {border-collapse:collapse;}

    /* This resolves the Outlook 07, 10, and Gmail td padding issue. Here's more info:
    http://www.ianhoar.com/2008/04/29/outlook-2007-borders-and-1px-padding-on-table-cells
    http://www.campaignmonitor.com/blog/post/3392/1px-borders-padding-on-table-cells-in-outlook-07 */

    /***** END BUG FIXES *****/
```

Optimizing for mobile devices

```
@media only screen and (max-device-width: 480px) {
```

```
/* Here you can include rules for the Android and iPhone native email clients. Device viewport dimensions are as follows:
```

```
    iPhone: 320px X 480px - portrait, 480px X 320px - landscape
```

```
    Android: 350px X 480px - portrait, 480 X 350 - landscape
```

```
    (These are average dimensions, the Android OS runs on several different devices) */
```

```
body[yahoo] #container1 {display:block !important}
```

```
/* example style */
```

```
body[yahoo] p {font-size: 10px}
```

```
/* example style */
```

```
/* You must use attribute selectors in your media queries to prevent Yahoo from rendering these styles. We added a yahoo attribute in the body tag to complete this fix.
```

```
http://www.campaignmonitor.com/blog/post/3457/media-query-issues-in-yahoo-mail-mobile-email/
```

```
http://www.emailonacid.com/blog/details/C13/stop\_yahoo\_mail\_from\_rendering\_your\_media\_queries
```

```
*/ }
```

```
@media only screen and (min-device-width: 768px) and (max-device-width: 1024px) {
```

```
/* Here you can include rules for the iPad native email client.
```

```
Device viewport dimensions in pixels:
```

```
    703 x 1024 - portrait
```

```
    1024 x 703 - landscape */
```

```
body[yahoo] #container1 {display:block !important}
```

```
/*example style*/
```

```
body[yahoo] p {font-size: 12px}
```

```
/*example style*/ }
```

Notes on embedded CSS:

1.) Be aware that Gmail will not read any of your embedded CSS

2.) Although we have seen the !important priority used in other examples, it is not necessary.

If you use "important" you can no longer overwrite your styles inline which is required for Gmail.

3.) The Android does not support "class" declarations outside of the media query. Here is more info on that:

http://www.emailonacid.com/blog/the_android_mail_app_and_css_class_declarations/

4.) You might want to consider duplicating your embedded CSS after the closing body tag for Yahoo! Mail in IE7 & 8.



Optimizing for mobile should be standard.
48% of emails are being viewed on mobile devices - Litmus

Coding your email

The best philosophy to embrace when designing and developing html emails for optimal deliverability is to use code simple enough that it will renders across all email clients and platforms, including those without full browser functionality. Declare as many parameters and attributes as possible, leaving nothing left for interpretation by various browsers, clients, devices and platforms, which do not adhere to definitive standards. Anticipate all configurations. Avoid JavaScript as most email software disables JavaScript by default.

Document Type Declaration (DTD)

Make sure to declare your DOCTYPE. DOCTYPE Declaration is the abbreviation for Document Type Declaration (DTD). The DOCTYPE Declaration (DTD or Document Type Declaration) serves two functions:

- (1) When validation your HTML code, it tells the HTML validator which version of (X)HTML standard the coding is supposed to comply with, and to validate against,
- (2) It tells the browser which standards compliant mode to render the code against. Declare your html email with the 4.01 Transitional doctype, as it allows elements and presentation attributes that should otherwise be included in the stylesheets to be included in line, which is crucial for HTML emails.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Failure to establish a doctype can lead to inconsistencies in the amount of margin that's added beneath <p> tags. By default, <p> tags have {margin: 1.12em 0;} applied according to the default HTML4 stylesheet, however browsers and email clients are not required to adhere to this, most notably RoadRunner and Yahoo. Failure to establish a doctype can also produce unexpected results when the border-collapse property is utilized (http://w3schools.com/Css/pr_tab_border-collapse.asp).

Content - Type and Character Set Declaration

Declaring a character set, tells the browser or email application how to interpret the text characters in the <body> of your email, especially with respect to special characters in non-Latin languages or when copying from a text editor like Microsoft Word. The most popular character sets are UTF-8 and ISO-8859-1. In website development, you can declare your character set employing a <meta> tag in the <head> of the document like this <meta http-equiv="Content-Type" content="text/html charset=UTF-8" />

Email clients however ignore the Content-Type defined within the <meta> tag. The Content-Type is automatically determined and set in the email's header by the server sending your email. This value can be changed but requires access to the email server. Contact your email service provider and ask them what Content-Type they set in the header when sending the emails if you desire. Alternatively convert all of your special characters to HTML entities to ensure consistent results. There are numerous online conversion tools and charts available to assist with converting characters. View the W3C HTML symbol entities reference chart for more details.

Laying-Out and Developing Your Email

HTML email layouts should always be developed using tables with inline styles as opposed to stylesheets (CSS). Avoid <div>s for layout work, tables will take more time and require more complexity, but ensure more consistent results and reduce the likelihood of the layout 'breaking'.

Murphy's Law:

Anything that can go wrong will go wrong...

So TEST, TEST, TEST!



Code like it's 1999!

Forget <div> Tags!

Forget <style> Tags!

TABLES ROCK!

Remember 640 x 480?
Web safe colors?
Parachute pants?

These are the tools of the HTML emails.

Forget what you know about Javascript, Frames, Forms, CSS, FLASH, AJAX... every acronym you spout out at cocktail parties. When it comes to HTML emails, "old school code" is the way to go.

Keep it simple... use TABLES, FONT tags, inline CSS (but be very aware that some email clients have limitations).

By keeping the code simple and the design straight-forward, you'll make life easier on the people who are receiving your message and it'll display more consistently.

-eROI

Tables

While using tables for layout may be a step back for those who like to code to the latest standards, sticking to tried and true techniques will ensure the most consistent rendering across the board. Using html tables also helps with problems caused by different email clients, browsers and devices that can interpret the same code differently.

Table-Based HTML Email Layout Tips

- Create one table each for the header, main column, and footer.
- Wrap these tables with another 'framing' table.
- Ensure the width of the 'framing' table does not exceed 98% to ensure proper rendering in clients such as Yahoo! Mail.

Table Tips:

- Set all possible **<table>** **<tr>** and **<td>** attributes with the tags to ensure maximum compatibility with clients and applications. Example: **<table border=0, valign=top, align=left, cellpadding=0, cellspacing=0>**.
- Declare all possible attributes, including dimensions (in pixels), font-attributes, and alignment, inline to avoid leaving anything up to interpretation by different email clients, browsers, and devices
- Set all possible attributes at the **<td>** element level as well.
- Background image support for **<table>** and **<td>** elements is inconsistent, even when it is supported images may not download for all recipients due to system and security settings for both known and unknown senders. If you decide to use background images, anticipate and develop a graceful contingency plan utilizing background colors and make sure to test the results to ensure the results are desirable.
- Avoid the use of **<colspan>** and **<rowspan>** tags, and try to minimize the number of nested tables (tables within tables) whenever possible. Older readers, including Lotus Notes have been known to have issues rendering these tags. Use the padding style to control margins within a **<td>** cell, margin style is inconsistency supported across email reader.

Cascading Styles Sheets (CSS)

Cascading Style Sheets are an area of debate among HTML email programmers. While it offers powerful formatting capabilities, support for internal and external CSS stylesheets varies across email clients, browsers and devices. Avoiding CSS ensures the most consistent and predictable results. Pre- knowledge of recipient rendering platforms can prove extremely valuable. As a rule, if CSS is utilized, include all styles specified in the stylesheet inline as a graceful fall - back. Campaign Monitor offers a current, and highly - detailed guide on CSS support among email clients and devices: [Campaign Monitor Guide to CSS Support in Email Clients](#).

Declaring a CSS Reset for Pixel Precise Work

Experienced web - designers utilize a CSS Reset to maximize control over layout and styling and ensure pixel - precise work. Resting the CSS resets removes all standardized formatting of HTML elements resetting all attributes to '0'. This allows for manual resetting of all HTML elements for maximum control of layout and prevents email clients, browsers and devices from interpreting attributes both declared and undeclared according to their native settings. To ensure consistency, the CSS should be reset by including both the follow CSS declaration in the stylesheet as well all manually setting all possible attributes on all referenced HTML tags.

Note: The following CSS Reset code includes numerous tags not supported by HTML email clients which will be ignored. The code is also useful for resetting the CSS for website design. To ensure optimal control for platforms which do not recognise stylesheets, ensure all possible attributes are declared inline for each HTML element utilized in the HTML.

CSS Reset Code (Include in the <body> not <head> tags for maximum compatibility)

```
<style type="text/css">
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr,
acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, dl, dt, dd,
ol, ul, li, fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
margin: 0;
padding: 0;
border: 0;
outline: 0;
font-weight: inherit;
font-style: inherit;
font-size: 100%;
font-family: inherit;
vertical-align: baseline;
}
/* remember to define focus styles! */
:focus {
outline: 0;
}
body {
line-height: 1;
color: black;
background: white;
}
ol, ul {
list-style: none;
}
/* tables still need 'cellspacing="0"' in the markup */
table {
border-collapse: separate;
border-spacing: 0;
}
caption, th, td {
text-align: left;
font-weight: normal;
}
blockquote:before, blockquote:after,
q:before, q:after {
content: "";
}
blockquote, q {
quotes: "" "";
}
</style>
```

Cascading Stylesheet Tips:

- Do rely upon CSS. Anticipate and code for a graceful fail.
- If you include CSS stylesheets include them within the <body></body> tags, not the <head></head> tags to prevent browser - based email applications from removing them by default. Do not use CSS shorthand, only basic CSS style declarations: instead of "font: 12px/16px Arial, Helvetica," specify each parameter separately: "font family, font-size, and line-height".
- Avoid using CSS or inline styles to replace bullets with images in unordered lists – Support is inconsistent.
- Some email servers strip out CSS lines that begin with periods such as class declaration. To remedy this add a space in front of any CSS that begins with a dot. (See below for example)

```
<style type="text/css">
.title {font-size:22px;}
</style>
```

- Ensure all styles declared at the <table> level are re - declared in the each contained <td> element as well. This can mean repetitive style declarations in multiple <td> cells.

Fonts

Font attributes can be set inline (directly to the HTML code) as well as in the CSS. For HTML emails it is recommended that font - attributes are set inline, and as many attributes as possible are set to prevent different bowsers and clients from treating the non - declared parameters differently. Some of the most common font attributes are:

Font Families

Utilize web - safe fonts. In order for a font to display correctly it must be installed on the recipient's computer, or supported by their email client or device, such as a blackberry. Web - safe fonts are generally shared across all browsers, clients, platforms, and devices, and thus safe to use without compatibility concerns.

When setting the font-family property, include several font names as a "fallback" system. If the browser does not support the first font, it tries the next font. Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available. If the name of a font family is more than one word, it must be in quotation marks, like fontfamily: "Times New Roman". More than one font family is specified in a comma - separated list. Typical web - safe fonts include Arial, Arial Black, Comic San MS, Courier New, Georgia, Impact, Tahoma, Times New Roman, Trebuchet MS, Verdana, Wingdings.

Font Size

There is much controversy and debate over which specification to use when setting font-size: Ems, Pixels, Points, and Percent. The majority of long - time designers agree upon specifying font size in pixels (px) not points (pt) or em (em). Different browsers, clients, devices and platforms interpret font sizes differently. A font may render and appear different when viewed on Internet Explorer on a PC, an iPhone, or a Mac. Setting font-size according to pixels allows for most precise control over fontsize, as it relates to pixels. Points and Ems can be resized according to the viewer's system settings. Be careful not to set the font to small and making the text difficult to read. A commonly used font-size is 12px.

iPhone Tip:

If you anticipate many iPhone recipients, be aware that the iPhone automatically sets a minimum fontsize of 13px, which can break your intended layout. This can be overridden by applying the CSS code below:

```
<style type="text/css">
div, p, a, li, td { -webkit-
text-size-adjust:none; }
</style>
```



Other font - attributes

font-weight: Regular, Bold, Italic, Underline, etc.

font-style: normal, italic

text-decoration: underline, overline, subscript, superscript

color: Black, #000000 (use common color names or hexadecimal code when possible)

line-height: Use pixels, not em or percentages

letter-spacing: Use pixels, not em or percentages

word-spacing: Use pixels, not em or percentages

Font Tip

Declare all possible font attributes, in all utilized and nested elements (<h1>, <h2>, etc., <table>, and <p>) to ensure maximum control and consistency over rendering.

Links

Links are a vital element of HTML emails. They direct your email's recipients to the offers and additional information presented in the email. Use the target="_blank" attribute to force links to open in a new browser and keep the original email open.

Link Tips

Some email clients overwrite your link colors with their defaults, and you can avoid this by taking two steps. First, set a default color for each link inline like so:

```
<a href="http://somesite.com/" style="color:#ff00ff">this is a link</a>
```

Next, add a redundant span inside the <a> tag

```
<a href="http://somesite.com/" style="color:#ff00ff"><span style="color:#ff00ff">this is a link</span></a>
```

To some this may be overkill, but if link color is important to your design then a superfluous span is the best way to achieve consistency.

Use the target="_blank" attribute for the HTML A tags so that people reading with a webmail service don't have the requested page appear within their webmail interface.

Images

Images are an essential part of email design. They are indispensable for conveying a message and creating visually effective and compelling emails, however they can present obstacles in terms of deliverability. **Not all email clients download and display images by default**, as a result of system and settings, as well as spam filters, resulting in delivery of an incomplete message.

To maximize deliverability and ensure your message is effectively conveyed, **avoid text-based images**, as any information conveyed through images will be lost when those image do not display or render as desired. To optimize your images for web delivery; use jpeg and .gif formats, avoid .png and .bmp images. **Keep your image sizes to a minimum**. Avoid using too many images to keep the overall size of the email small (< 100KB). **Include the image description and details in the <alt> tags to describe your images, and assign them hyperlinks**. Use images of the desired size whenever possible, and set the height and width attributes for each image to prevent the layout from breaking when an image does not load and carries no size specifications.

Make sure the images are consistent with your corporate image and the message you are trying to convey.

Image Tips

- Declare each image's dimensions, height and width, in pixels.
- Declaring alt parameters for all images. This helps convey the message of an image when fails to rendering and ensure the message is still conveyed.
- For layout considerations, avoid using large images above the fold in the email. This is another classic spammer practice and can increase the likelihood an email will be tagged as spam.
- Use only .GIF or .JPG formats for all images, even if it means some additional file size. Avoid animated .gifs files. Support is inconsistent.
- Avoid .PNG images. Lotus Notes 6 and 7 does not support 8-bit or 24-bit PNG images and can result in rendering errors, most commonly red 'X's.
- Close a <td> on the same line as an image, do not insert a line-break. Avoid using 1×1 pixel images to force precise spacing, this can trigger spam filters.

Client Specific Image Tips

Hotmail: Windows Live Hotmail adds additional pixels of padding below images. Apply the following style to images to prevent this occurring:

Inline:

```

```

CSS:

```
<style type="text/css">
```

```
img {display:block}
```

```
</style>
```

Outlook: Outlook 2007 does not recognize the <alt> parameter.

Text

The copy is generally the most important element of an email. Reliable rendering and formatting of text is crucial for conveying a strong, professional message and presentation. Avoid copying or importing copy and content from applications, websites, or word processors (such as Microsoft Office) as they can generate superfluous code inconsistent with strict HTML standards.

Experienced HTML coders work exclusively with HTML source code and are aware of these inclusions. When concerned, pasting and re-copying your text for a simple text editor (such as Microsoft Notepad) will remove all text formatting.

Text Tips

- Use carriage returns (Shift-Enter) instead of line-breaks (Enter) to ensure consistent line spacing and rendering. In HTML source code:

 instead of <p>
- Avoid inputting symbols directly, use html character codes. See Character Set Declarations above Coding Conclusion and Best Practices
- Whenever possible, use the simplest solution possible, it will ensure the greatest consistency in rendering.
- Declare all possible attributes. It is better to be thorough and ensure everything is declared than leaving attributes up to interpretation by the recipients applicant or device.
- Avoid CSS, if you use it, anticipate a graceful fall back for when it fails to load.
- Avoid JavaScript. Most email software disables JavaScript.

Avoiding the Spam Filter

Use Spam - Friendly Subject Lines

Subject lines are particularly scrutinized by spam filters. Avoid the following scenarios in your subject lines:

- Too many capital letters compared to lowercase letters.
- Repeating Capital Letters in a row.
- Gaps In Between Letters. (ex. s*p*a*m)
- Repeting letters and/or symbols.
- Special Characters. Overusing special characters like \$, #, %, etc.
- Punctuation. Overusing punctuation marks.
- Word/Space Ratio. Using too many blank spaces.
- First Character Is Special. Starting your subject line with a special character.
- SPAM filter - triggering words. (see below).

Avoid spam words and phrases

Some words and phrases hold more significance for spam filter triggers than others. While it's sometimes necessary to use these, refrain from using them whenever possible. [Here is a list of words](#) and phrases that can trigger spam filters.

Image - Related Spam Filter Triggers

- Image to Text Ratio. Emails that are image heavy, or have large images, especially above the fold, can get flagged as SPAM. Image to text ratio should be 1 image per 1 paragraph of text.
- Spacer images. Avoid using 1x1 pixel images for precision spacing and layout, they can trigger SPAM filters.

CAN-SPAM Act: [10 Point Checklist](#)

Thank you to the cited sources!

This document was created/compiled with respect to the original sources and all of the sources have been noted. The purpose of this document is to act as an educational resource/learning aid. It is not intended for sale or resale. All cited portions remain the intellectual property of the respected authors.

Please view the original documents:

Email on Acid (boilerplate – [eMailology](#))

C&A2011 Email Guide (majority of body text – [full guide](#))

EE12 - Do It - McAfee - HTML Email Design – (1999 header - [full guide](#))

eROI – Email Design and Style Guide (1999 Column – [full guide](#))

Definitive Results is dedicated to driving your company's marketing success.



Definitive Results