Best Practices for SAP e-mailing

How to generate beautiful HTML e-mails from SAP®

Chris Scott

Arch

Best Practices for SAP e-mailing

How to generate beautiful HTML output from SAP using Floe

1st Edition, February 2016

References

I am grateful to the following on-line resources from which I have borrowed significantly:

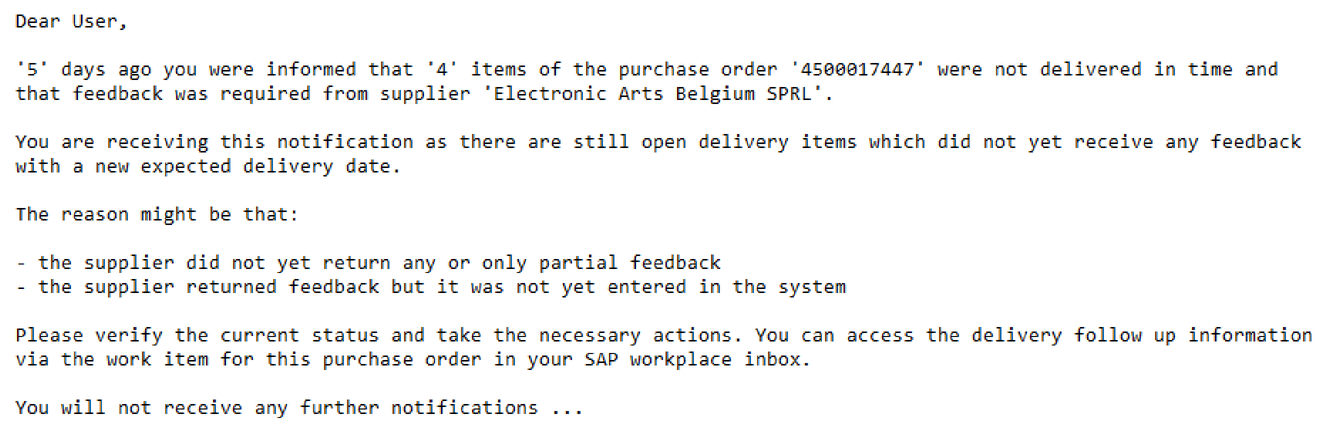
- http://www.sitepoint.com/how-to-code-html-email-newsletters/
- https://www.campaignmonitor.com/blog/post/3317/correct-doctype-to-use-in-html-email/
- https://sendgrid.com/blog/embedding-images-emails-facts/
- https://support.sendgrid.com/hc/en-us/articles/200184928-HTML-Rendering-The-Do-s-and-Dont-s-of-Cross-Platform-Email-Design
- http://sixrevisions.com/web_design/creating-html-emails/
- http://code.tutsplus.com/tutorials/getting-started-with-html-emails--net-17200
- Outlook 2007, 2010 & 2013 and their infamous Word Rendering Engines (ebook by *Email on Acid*)

Table of Contents

Chapter 1: Introduction

E-mail is an important method of communication for almost every business. It is used in sales for order confirmations, delivery notifications and invoicing. It is used in purchasing to send purchase orders and remittance advice notifications. It is used for sending newsletters to customers, contracts to suppliers, and many kinds of employee correspondence. E-mail is used widely in marketing, for event and webinar invitations, to drive website traffic and lead generation. The public sector use e-mail for citizen communication. Banks and Financial Institutions use e-mail to notify customers of rate changes, account statements, quotations and renewal notices. Internally within organisations, notifications of work to be completed, or notifications that work has been completed, are often essential to the smooth running of key business processes. E-mail is used for travel documentation; ticket booking; parking receipts; the list goes on and on.

SAP core systems have long had the capability to create and send e-mail. However, in general the type of e-mail generated is either a plain-text notification, or a simple e-mail with a PDF attachment. The output design technologies (SAPScript, SMARTFORMS and SAP Interactive Forms) were designed primarily for printed output, and are not capable of rendering e-mail content. Instead, a PDF document is created and sent as an attachment to a fairly basic e-mail. This approach has some issues: PDF rendering can be CPU-intensive, and PDF document sizes can be large. PDF attachments can clog up message servers or get rejected by firewall rules.



```
Dear User,

'5' days ago you were informed that '4' items of the purchase order '4500017447' were not delivered in time and
that feedback was required from supplier 'Electronic Arts Belgium SPRL'.

You are receiving this notification as there are still open delivery items which did not yet receive any feedback
with a new expected delivery date.

The reason might be that:

- the supplier did not yet return any or only partial feedback
- the supplier returned feedback but it was not yet entered in the system

Please verify the current status and take the necessary actions. You can access the delivery follow up information
via the work item for this purchase order in your SAP workplace inbox.

You will not receive any further notifications ...
```

Typical SAP e-mail without formatting

As e-mail recipients, we expect a level of user experience well beyond the type of static, plain e-mail normally generated from SAP. We expect e-mails with rich content; colours, pictures and logos; with corporate branding; with all the detail we require displayed concisely and elegantly, and most importantly, we expect details of the business transaction within the body of the e-mail rather than in a separate attachment.

In 2014 we challenged ourselves at Arch to build a better way to generate SAP e-mails that could be used in many different scenarios and produce e-mails that looked great on any device, through any browser. This means working with HTML e-mail content, and dynamically building content at run-time. We launched Floe in 2015 as a low-cost development tool for exactly this purpose.

Awesome SAP e-mail made easy
Create and manage beautiful e-mail templates for all your automated SAP correspondence
1
2
3

It turned out that Floe is only half of the solution: Anyone designing more graphical HTML e-mail content soon discovers that:

i) You cannot use many common HTML approaches like CSS and scripting, and so a particular approach for HTML e-mail design is required.

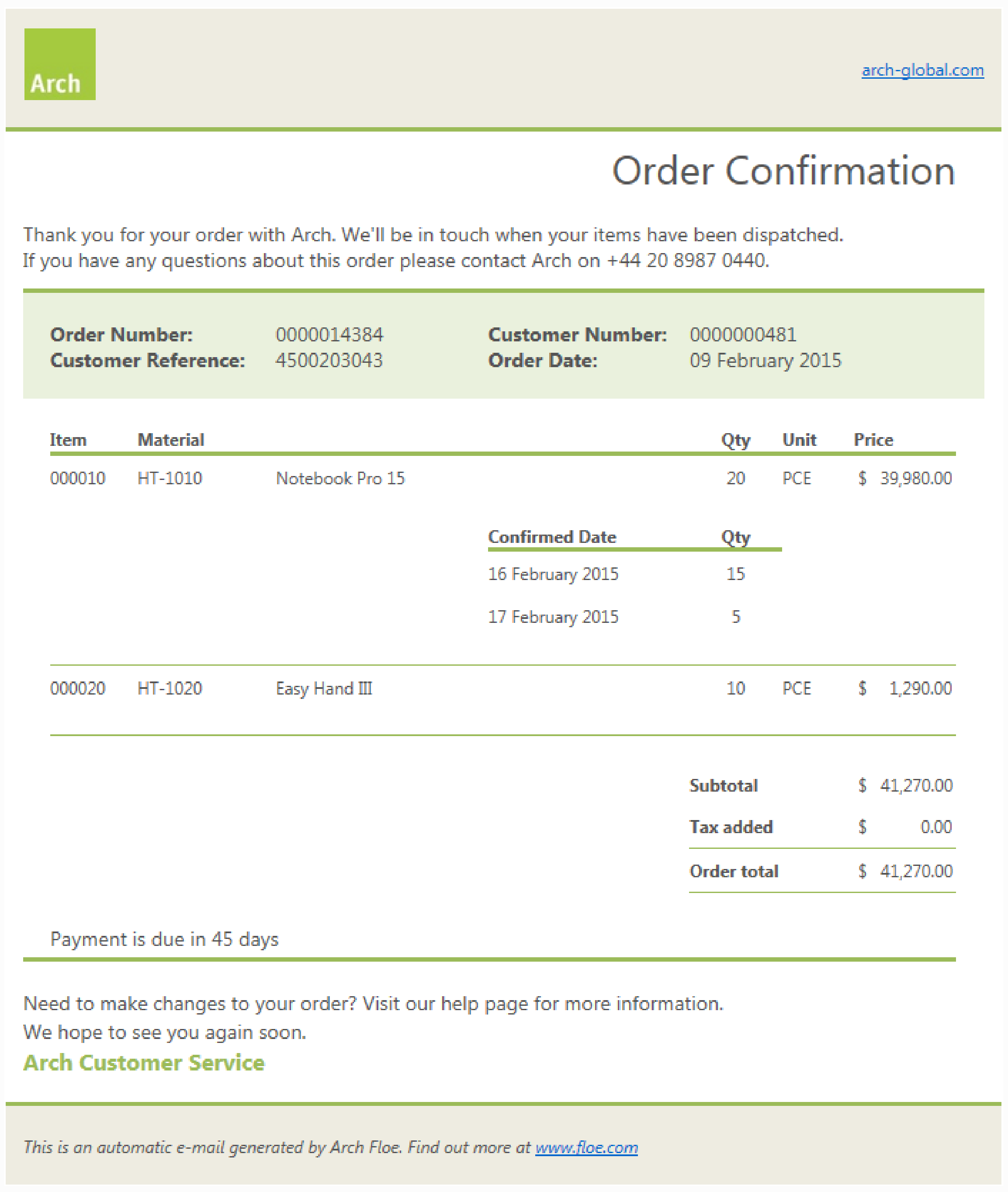
ii) Different e-mail clients interpret HTML e-mail content in different ways, and there are many different tips and tricks required to create content that looks fantastic on all the different e-mail clients. [The biggest culprit is Microsoft Outlook, and the biggest challenge is designing for Microsoft Outlook!]

This ebook is not a Floe brochure or a Floe configuration guide. (These can be found on the Arch website and Arch support site respectively.) Nor is this book intended to be a tutorial for building HTML e-mails.

The purpose of this ebook is two-fold: First to introduce Floe functionality and the recommended approach and techniques to produce best results from the product. And second, to bring together recommendations and guidelines for designing beautiful e-mail content for any device. This ebook should be an invaluable reference guide for any developer tasked with the creation of better e-mails from SAP.

Chapter 2: Introducing Floe

Floe is an add-in for SAP which provides the functionality to manage HTML e-mail templates, render e-mails dynamically based on SAP data and business logic, and distribute those e-mails using the SAP Business Communication Service (BCS). Floe is a developer tool and in most scenarios it is invoked by a function module call that is embedded within custom programs. In this way Floe can be used for any SAP e-mail communication, and can be incorporated into existing business processes easily.



Arch

arch-global.com

Order Confirmation

Thank you for your order with Arch. We'll be in touch when your items have been dispatched.
If you have any questions about this order please contact Arch on +44 20 8987 0440.

Order Number:	0000014384	**Customer Number:**	0000000481
Customer Reference:	4500203043	**Order Date:**	09 February 2015

Item	Material		Qty	Unit	Price
000010	HT-1010	Notebook Pro 15	20	PCE	$ 39,980.00
000020	HT-1020	Easy Hand III	10	PCE	$ 1,290.00

Confirmed Date	Qty
16 February 2015	15
17 February 2015	5

Subtotal	$ 41,270.00
Tax added	$ 0.00
Order total	$ 41,270.00

Payment is due in 45 days

Need to make changes to your order? Visit our help page for more information.
We hope to see you again soon.
Arch Customer Service

This is an automatic e-mail generated by Arch Floe. Find out more at www.floe.com

Example sales order confirmation e-mail showing nested, repeating sections

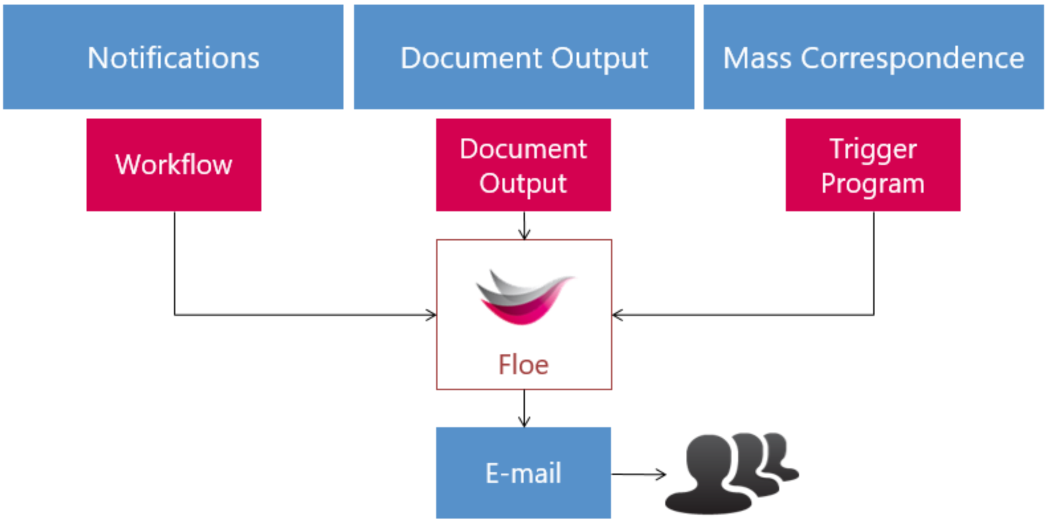
At design-time, Floe is a 'SAP module', accessed through the SAP Implementation Guide (IMG), that enables the configuration of e-mail types and associated configuration using HTML e-mail templates or template fragments ('blocks'). Business logic can be added through customer exits for the derivation of data, fragments, attachments, images and recipients.

E-mail templates can include repeating sections or rows to include table-based information. Floe also supports nested repeating sections in order to support document output such as sales order item schedule lines.

When SAP document output is converted from PDF to HTML e-mail, the resulting e-mail is greatly reduced in size, which can provide a range of related benefits in terms of performance, database size and mail server processing.

Floe has functionality for e-mail recipient list management and mass e-mail processing, whereby an e-mail can be sent to a large number of recipients. A crucial difference between Floe and third-party e-mail tools (such as Mailchimp) is the degree of tailoring possible: Third-party tools are capable of personalisation through simple variable substitution, but Floe adds a greater level of sophistication.

With Floe you can alter the e-mail content based on business logic: For example, you could include or exclude sections of the e-mail based on recipient preferences or other data within SAP like supplier country, customer group, personnel area, etc. A single e-mail template may be used to create many different e-mails, with different e-mail content such as paragraphs, links, images and attachments, based on each recipient.



Floe can be triggered from a variety of SAP processes

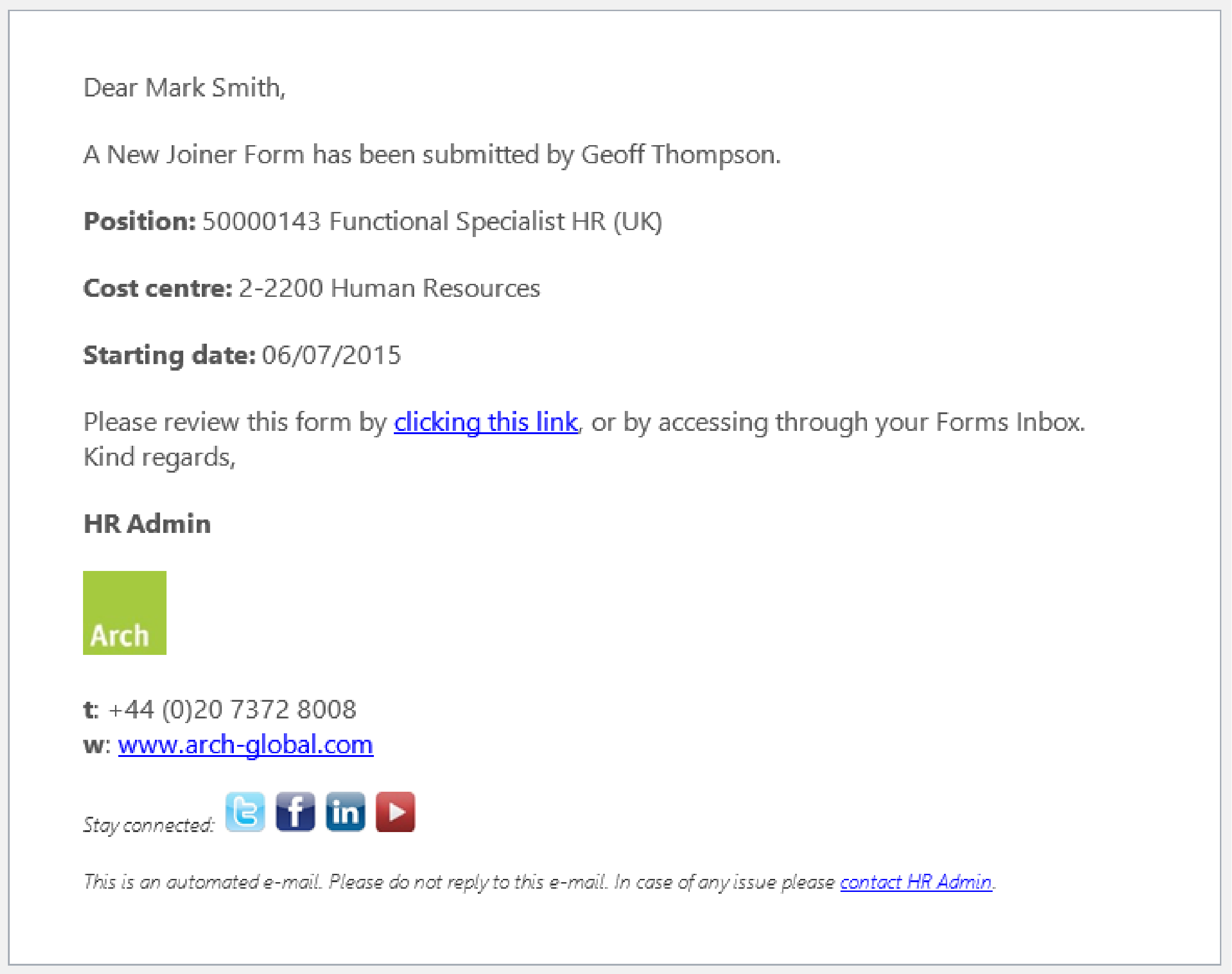
Floe could be used by every single SAP customer to improve every single e-mail generated by the SAP system. Apart from simple ABAP® skills, the only other required skill is HTML. Since there are no separate CSS or JavasScript components, the level of HTML skill required to build e-mail output is much lower than the skills required to build websites.

Chapter 3: Design Approach

A different design approach is required based on the business scenario the e-mail will support, because different scenarios are likely to require different levels of corporate branding, or need to be designed for more devices/e-mail clients, or include more graphical elements.

Internal notifications

Internal notifications are likely to be less reliant on graphics, and in general will not be viewed outside the enterprise, so a more simple e-mail template is more appropriate. Simpler templates are much easier to maintain, and so you should take the opportunity to keep the design simple where possible.



Dear Mark Smith,

A New Joiner Form has been submitted by Geoff Thompson.

Position: 50000143 Functional Specialist HR (UK)

Cost centre: 2-2200 Human Resources

Starting date: 06/07/2015

Please review this form by clicking this link, or by accessing through your Forms Inbox.
Kind regards,

HR Admin

t: +44 (0)20 7372 8008
w: www.arch-global.com

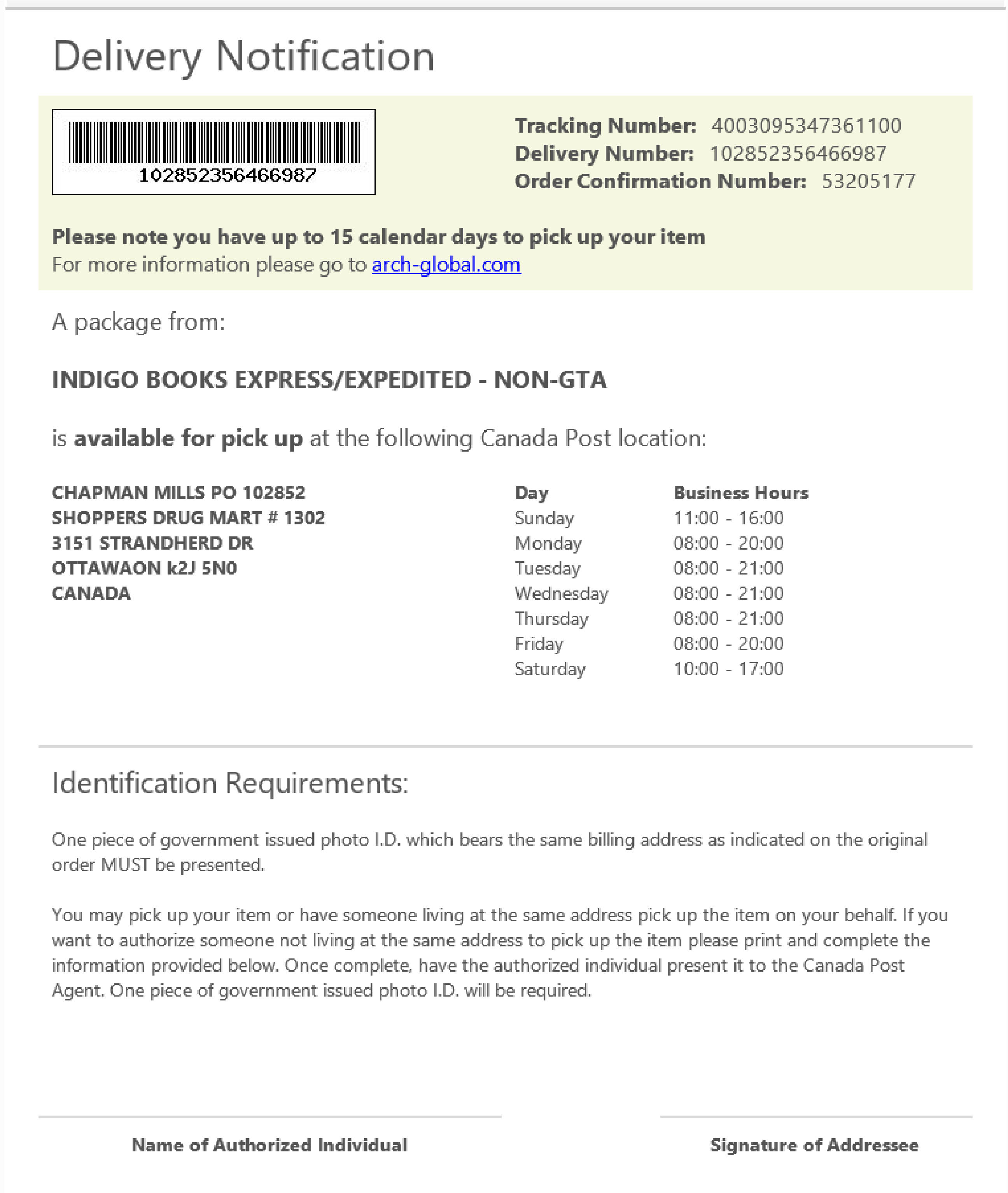
Stay connected:

This is an automated e-mail. Please do not reply to this e-mail. In case of any issue please contact HR Admin.

Simple SAP notification e-mail

External notifications & SAP document output

Where the e-mail is designed for external consumption, or where it contains a lot of SAP data, then a more complex template will be required, with more graphical elements, requirements to align content horizontally and vertically, and be supported on a wide range of e-mail clients. This type of requirement may be met using template fragments that are combined at run-time.



Delivery Notification

102852356466987

Tracking Number: 4003095347361100
Delivery Number: 102852356466987
Order Confirmation Number: 53205177

Please note you have up to 15 calendar days to pick up your item
For more information please go to arch-global.com

A package from:

INDIGO BOOKS EXPRESS/EXPEDITED - NON-GTA

is **available for pick up** at the following Canada Post location:

CHAPMAN MILLS PO 102852
SHOPPERS DRUG MART # 1302
3151 STRANDHERD DR
OTTAWAON k2J 5N0
CANADA

Day	**Business Hours**
Sunday	11:00 - 16:00
Monday	08:00 - 20:00
Tuesday	08:00 - 21:00
Wednesday	08:00 - 21:00
Thursday	08:00 - 21:00
Friday	08:00 - 20:00
Saturday	10:00 - 17:00

Identification Requirements:

One piece of government issued photo I.D. which bears the same billing address as indicated on the original order MUST be presented.

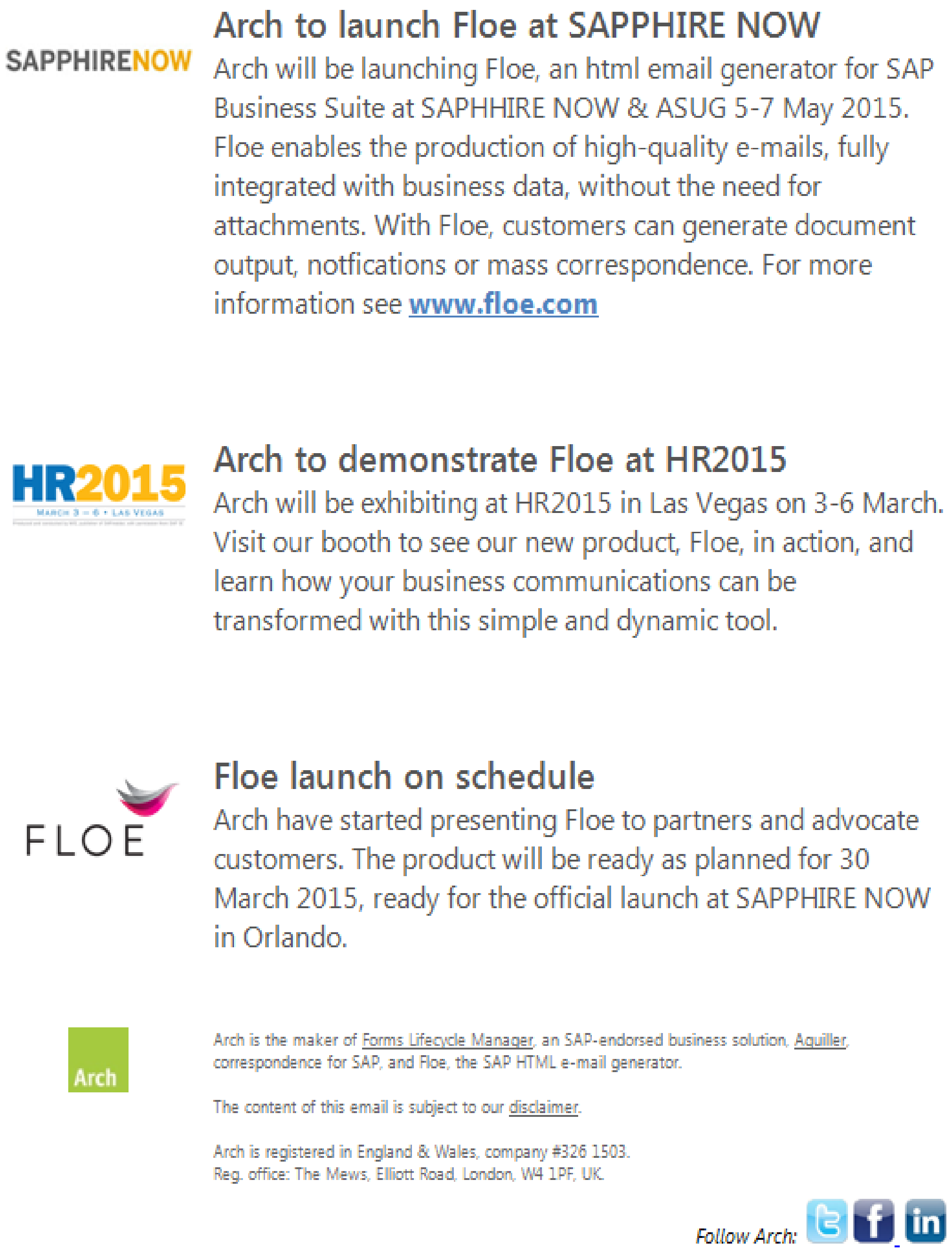
You may pick up your item or have someone living at the same address pick up the item on your behalf. If you want to authorize someone not living at the same address to pick up the item please print and complete the information provided below. Once complete, have the authorized individual present it to the Canada Post Agent. One piece of government issued photo I.D. will be required.

Name of Authorized Individual **Signature of Addressee**

External notification driven by SAP document output

Mass Correspondence

For mass correspondence scenarios the approach will depend on the degree of personalisation required. Where the same e-mail body content is sent to all recipients with simple variable substitution, then a single template can be used. This is likely to have some complexity but be presented in discrete sections, which are easy to maintain and replicate. In scenarios where the content is highly personalised, the template will be designed in fragments in order for logic to be added to include or exclude a fragment dynamically.



Arch to launch Floe at SAPPHIRE NOW

Arch will be launching Floe, an html email generator for SAP Business Suite at SAPHHIRE NOW & ASUG 5-7 May 2015. Floe enables the production of high-quality e-mails, fully integrated with business data, without the need for attachments. With Floe, customers can generate document output, notfications or mass correspondence. For more information see **www.floe.com**

Arch to demonstrate Floe at HR2015

Arch will be exhibiting at HR2015 in Las Vegas on 3-6 March. Visit our booth to see our new product, Floe, in action, and learn how your business communications can be transformed with this simple and dynamic tool.

Floe launch on schedule

Arch have started presenting Floe to partners and advocate customers. The product will be ready as planned for 30 March 2015, ready for the official launch at SAPPHIRE NOW in Orlando.

Arch is the maker of Forms Lifecycle Manager, an SAP-endorsed business solution, Aquiller, correspondence for SAP, and Floe, the SAP HTML e-mail generator.

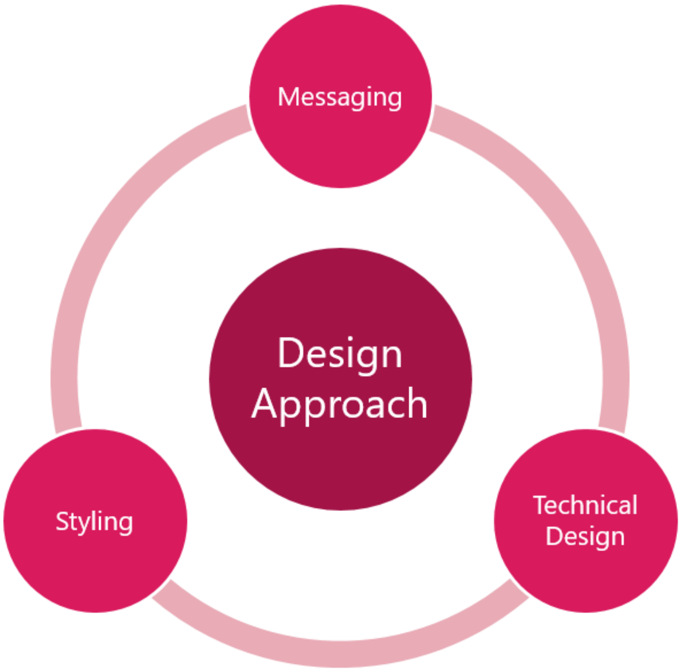
The content of this email is subject to our disclaimer.

Arch is registered in England & Wales, company #326 1503.
Reg. office: The Mews, Elliott Road, London, W4 1PF, UK.

Mass correspondence with little personalisation: single template

Blueprinting

When blueprinting the e-mail, you should consider the messaging, the styling and the technical design.



Design approach: Messaging, Styling and Technical Design

Messaging

When designing the e-mail firstly consider the content of the messaging. Is the e-mail for internal employees; for business partners; for customers; for prospects? Is there some marketing content to be added, and will this content be static or change over time?

What is the purpose of the e-mail? Is it a 'call to action' to register for an event, to drive website traffic, to take advantage of a sales promotion, to renew a service? Or is it an instruction for an employee to conduct one or more tasks, to approve a request or resubmit a form? Or is the e-mail a notification that a service has happened or will happen, with no action required by the recipient?

Where action is required by the recipient we need to ensure that the action is clear, and not lost within a large e-mail. For notifications, we need to include content pertinent for the recipient and exclude everything else: The e-mail needs to be concise and precise.

Consider the degree of personalisation required: Typically we know who the e-mail is being sent to so that we add the recipient's name easily. But is there other content that can or should be tailored, or included/excluded based on the recipient. What business rules should determine this?

Are you expecting the recipient to be able to reply to the e-mail? Consider where the e-mail is sent from, and whether/how the inbox of the sender is monitored.

Styling

Separately from the messaging of the e-mail content, consider the styling. Do you have internal communication and branding standards to follow? Do you have a colour palette that should be used and logos to be included?

In general, styling decisions will not be specific to a single e-mail, and so whatever style you select for the first e-mail is likely to be used for subsequent e-mail templates: You should arrive at your final design purposefully, not by accident.

Note that in many circumstances, the font you use will be over-ridden by default device fonts or e-mail client settings made by the recipient, so don't get too hung up about replicating corporate fonts, as the time may be largely wasted.

Design your e-mail so that the most important content appears near the top of the e-mail so that it is visible immediately when opened by a recipient.

Technical Design

Once the message and styling requirements are defined, you can focus on the technical design. As part of the technical design you should consider:

i) How to design the HTML tables and fragments:
   a. How many columns are required in order to align different sections of the body;
   b. How many sections will make up the e-mail;
   c. Whether any sections are replicated in other e-mails templates;
   d. Whether any sections need to repeat.

ii) How images are going to be included (where the images will be stored).

iii) Whether any attachments will be included with the e-mail, and where those attachments will be stored (or how they will be generated).

iv) What data from SAP is going to be included within the e-mail body and subject: define variables for each discrete piece of data.

v) Whether any formatting logic is required to SAP raw data (such as currencies and dates) before substitution.

vi) Whether any SAP logic is required to determine any e-mail content.

Chapter 4: Coding HTML E-mails

Every e-mail tool converts the HTML content in different ways, and so there can be a big difference in results. The challenge is to build an e-mail template that is maintainable and offers a wide range of support to the most commonly used e-mail tools.

Check-out pre-built templates from *Mailchimp* and/or *Campaign Monitor*

Mailchimp and Campaign Monitor are cloud-based solutions for mass e-mailing, and they both offer a number of free templates on which to build your own designs. The templates offered by these companies are designed for a wide range of e-mail clients, and so they are a great starting place: You can sign-up for free, download a template and then add your own content.

E-mail design differs from web page design

E-mail design using HTML relies on HTML tables and inline CSS. In general the e-mail should be narrower: 600px has become a standard, and this restricts the number of columns & horizontal space available. In general, e-mails which can wrap and look good in a very narrow window must be very simple, with few images.

```
<body>
  <table width="100%" cellpadding="0" cellspacing="0" bgcolor="e4e4e4">
    <tr>
      <td>
        <table id="header_table" cellpadding="20" cellspacing="0" width="600"
align="center">
        </table>
        <table id="main_table" cellpadding="0" cellspacing="15" bgcolor="ffffff"
width="600" align="center">
         <tr>
           <td>
            <table id="header" cellpadding="10" cellspacing="0" align="center">
              <tr>
                <td width="570" bgcolor="7aa7e9"><h1>Newsletter</h1></td>
              </tr>
            </table>
           </td>
         </tr>
         <tr>
           <td><p> More content here.</p></td>
         </tr>
       </table>
       <table id="bottom_table" cellpadding="20" cellspacing="0" width="600"
align="center">
       </table>
      </tr>
    </td>
  </table>
</body>
```

Design for Outlook first

Since many of the restrictions result from the Microsoft Outlook rendering engine, I find it better to design for Outlook first, and then test on other devices and clients, adding in styling for the other e-mail tools that Outlook will ignore. It's an iterative process unless you are using a pre-delivered template.

Scaling for smaller devices

Use **@media** definitions to add responsive capability to the e-mail design.

For example, in the following code snippet, you would need to assign the class 'maintable' to the highest-level table in the e-mail output.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>
      Sample e-mail template
    </title>
    <style type="text/css">

      /* MOBILE STYLES */
      @media only screen{
        /* If media queries are supported, make all images fluid. */
        img{
          width:100%;
        }
      }

      @media only screen and (max-width:480px){
        /* Force iOS Mail to render the email at full width. */
        body{
          width:100% !important;
          min-width:100% !important;
        }

        /* Set width of maintable */
        .maintable{
           Clear: both;
           width:95% !important;
           font-size:14px !important;
        }

      }
    </style>
  </head>
```

Defining HTML tables

Once you have a conceptual design, you will be able to define the entire e-mail content as a series of tables. Simple notifications may only require 2 columns for most of the content, and complex document output may require over 10 columns.

Typically, a container table is used to store the entire e-mail body, and then nested tables are used to store the various sections of the e-mail. Sometimes better results can be achieved by setting the container table width to 98% or 95%.

```
<body>
    <center>
      <table align="center" border="0" cellpadding="0" cellspacing="0"
height="600px" width="100%" id="maintable">
        <tr>
          <td align="center" valign="top">
            <table align="center" border="0" cellspacing="0" cellpadding="0"
width="100%" id="header_section">
              ...
            <table align="center" border="0" cellspacing="0" cellpadding="0"
width="100%" id="body_section">
              ...
            <table align="center" border="0" cellspacing="0" cellpadding="0"
width="100%" id="footer_section">
              ...
          </td>
        </tr>
      </table>
    </center>
</body>
```

In this example, the e-mail is split into 3 sections, header_section, body_section and footer_section. The body section would be split into further tables to closely control the content. This is to support:

i) Horizontal and vertical positioning;

ii) Content that can change position on smaller devices.

Lists

Use tables for lists, to have better control of the text wrapping and alignment. In general, for any important positioning of content it is better to use a nested table instead of using CSS properties 'margin' or 'padding', since these are interpreted differently by different e-mail tools.

```
<table cellpadding="2" cellspacing="2" border="0">
      <tr>
            <td valign="top">1.</td>
            <td>The first item in my list.</td>
      </tr>
      <tr>
            <td valign="top">2.</td>
            <td>The second item in my list.</td>
      </tr>
</table>
```

Formatting tables: common controls

cellpadding	On the <table> tag define the space between the cell wall and the content using cellpadding (not 'padding'). The 'padding' property will work in an individual <td> but not the 'margin' property.
cellspacing	On the <table> tag define the space between the cells using cellspacing.
border	In general the border should be hidden `border="0"`
align	Use to align cell content to 'left', 'right' or 'center'.
valign	Use to align cell content to 'top', 'middle' or 'bottom'.

border-collapse	This is required to remove a white border added by Outlook if you are using a background colour. It can be set in the styles declaration or inline:
	`<style type="text/css">`
	`      table{border-collapse:collapse;}`
	`</style>`
	Or
	`<td style="border-collapse: collapse;">... </td>`

width	In general work with percentages, unless the design forces you to use absolute widths. It is recommended to define widths within the individual table cell on the top row of the table.
colspan	Use to span content over table columns within a row. `<td colspan="2" </td>`
line-height	Set rows to a particular height using line-height with mso-line-height-rule, since otherwise Outlook ignores the 'line-height' property: `<td style="mso-line-height-rule:exactly; line-height:50px;">` Note that Outlook sets a 2-pixel minimum height for table rows.

Text boundaries in Outlook

Some versions of Outlook automatically add in breaks in the content for long tables. If this causes an issue then split up the design into more, smaller tables.

Removing unwanted gaps between tables in Outlook

Sometimes Outlook will add in gaps between tables, and the following example (reproduced from *Email on Acid*) shows how this can be overcome using mso-table-lspace and mso-table-rspace. This issue occurs when you are using left/right align in your table.

```
<style type="text/css">
      table {border-collapse: collapse;}
</style>
<table border="0" width="600" cellspacing="0" cellpadding="0">
      <tr>
            <td>
                  <table align="left" width="198" border="0" cellpadding="0" cellspacing="0" style="border:1px solid #00CC99">
                        <tr>
                              <td bgcolor="#00CC99"><p style="mso-table-lspace:0;mso-table-rspace:0;">Content in 1</p></td>
                        </tr>
                        <tr>
                              <td bgcolor="#00CC99">Content in 1</td>
                        </tr>
                  </table>
                  <table align="left" width="198" border="0" cellpadding="0" cellspacing="0" style="border:1px solid #33FFFF">
                        <tr>
                              <td bgcolor="#33FFFF"><p style="mso-table-lspace:0;mso-table-rspace:0;">Content in 2</p></td>
                        </tr>
                        <tr>
                              <td bgcolor="#33FFFF">Content in 2</td>
                        </tr>
                  </table>
                  <table align="left" width="198" border="0" cellpadding="0" cellspacing="0" style="border:1px solid #993366">
                        <tr>
                              <td bgcolor="#993366"><p style="mso-table-lspace:0;mso-table-rspace:0;">Content in 3</p></td>
                        </tr>
                        <tr>
                              <td bgcolor="#993366">Content in 3</td>
                        </tr>
                  </table>
            </td>
      </tr>
</table>
```

Changing the table properties for Outlook and Internet Explorer

Where table properties seem to work for Outlook and nothing else, or vice versa, you can add logic to define the table properties based on the e-mail viewer.

```
<!--[if (mso)|(IE)]>
<table class="main" style="width:90%; " border="0" cellspacing="0"
cellpadding="10"
align="center">
<!--<![endif]-->
<!--[if (!mso)&(!IE)]><!-->
<table class="main" style="width:90%; " border="0" cellspacing="0"
cellpadding="0" align="center">
<!--<![endif]-->
```

Similarly, you can add styling specifically for Outlook:

```
<!--[if gte mso 9]>
    <style type="text/css">
    /* Your Outlook-specific CSS goes here. */
    </style>
<![endif]-->
```

Adding text styling with inline CSS

Once the (nested) table layout is in place, styling is added to text using inline styling. This includes properties such as font types, sizes, weights, font colours and background colours. This is a repetitive task, since every element needs to have individual style definitions. Since styles are not defined globally there is little benefit of using <h1>, <h2> and <div> definitions; instead, you can use <p> tags throughout.

```
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;
font-weight:bold;font-size:14px;">Incident link:</p>
```

For clarity: you cannot specify styles in the <head> section, and you cannot reference external stylesheets. Outlook does not support any inheritance, and so you have to define the styles at the lowest level; the individual HTML element.

CSS shorthand is also not supported, and so for example:

```
<p style="font: 12px Arial, Helvetica; color:#FFF;">Company:</p>
```

Needs to be changed to:

```
<p style="font-family: Arial, Helvetica; font-size:12px;
color:#FFFFFF">Company:</p>
```

You should avoid using <div> because of various e-mail tool support issues, instead use html tables to set padding etc. Outlook will often convert div and heading tags to paragraphs.

Styling can be set on a <span> element.

If you already have designed e-mail content with embedded CSS and you need to convert it to inline CSS, then check out http://premailer.dialect.ca/. This is an on-line tool for this automatic conversion.

Working with images

There are a few simple guidelines to follow when working with images:

- *Web server.* Rather than embedding images, it is best practice is to store the image on a web server, in a directory where it won't be deleted.
- *File type.* The file type of each image should be .jpg or .gif because some e-mail clients do not support .png images.
- *Alternate text.* Specify an 'alt' attribute for each image, since many e-mail clients do not show the images by default, but will display the text defined in the 'alt' property instead. For the same reason, images should not contain important text, because many recipients may not elect to view the images. As a general rule, images should include graphical content only.
- *Height and width.*
  - Always define the height and width attributes for every image, in order to maintain the e-mail layout when the images are turned off, and improve results in some e-mail clients.
  - All graphics should be saved with the correct dimensions, as stretched images (where the height and width defined in the HTML does not match the image properties) may not render correctly.
  - Set the cell height and width to match the height and width of the image for better results in Outlook.

```
<td width="600" height="80">
<img width="600" height="80"
  src="http://www.mywebsite.com/images/my_image.jpg" />
</td>
```

- *Header images.* Don't put large images at the top of the e-mail as this will mean that the email text content will not be seen when the e-mail is in a preview pane. Restrict the height of images at the top of the e-mail.
- *Display:block.* Use the display:block property in your embedded CSS. This removes unwanted spacing in some e-mail clients.

```
<style type="text/css">
     table {border-collapse: collapse;}
     img {display:block;}
</style>
```

- *Closing table cells.* Do not close a <td> cell immediately below the <img> tag as this can add unwanted spacing. Instead put the </td> immediately after the <img> tag on the same line.
- *Margin/padding.* Do not use margin or padding properties within the <img> tag as this is not supported by Outlook. Instead, control this in the table cell.
- *Float tags.* Avoid 'float' tags. Instead, use 'align'.

```
<img width="600" height="80" align="right"
 src="http://www.mywebsite.com/images/my_image.jpg" />
```

- *Large images.* Avoid images of height greater then 1728px, as these will get cropped in Outlook.
- *Background images.* Avoid background images if you can, as these are very complicated to get working in Outlook, and troublesome in other e-mail clients.

Miscellaneous other guidelines

Links

- If you want a different colour than the default, you can use a <span> around the link text.

```
<p style="color:#000000">
   <a href="link" target="_blank">
      <span style="color:#000000">link</span>
   </a>
</p>
```

- Alternatively you can embed CSS to style links:

```
<style type="text/css">
      a.archLink{
        font-family:Segoe UI; color:#4F81BD;
      }
<style>
```

- Add target="_blank" to each <a> tag so that links are opened in a new window for recipients using web mail.

Background colours

- If you need to set a background colour, do this in the table cell properties with the 'bgcolor' attribute.

DOCTYPE

- Adding a DOCTYPE definition seems to make little difference in most scenarios. However, Campaign Monitor recommends the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional //EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

JavaScript

- Do not use any JavaScript as in most cases it will not work.

Text wrapping

- Where a block of text contains no spaces it does not wrap properly in Outlook. To enforce the wrap within the width of the cell, add:

```
<td style="word-break:break-all;">
    …
</td>
```

Animated GIFs

- Do not use animated gifs as these are not supported by Outlook.

BR / HR

- Avoid <br> and <hr> tags; use tables instead.

1 x 1 pixel images

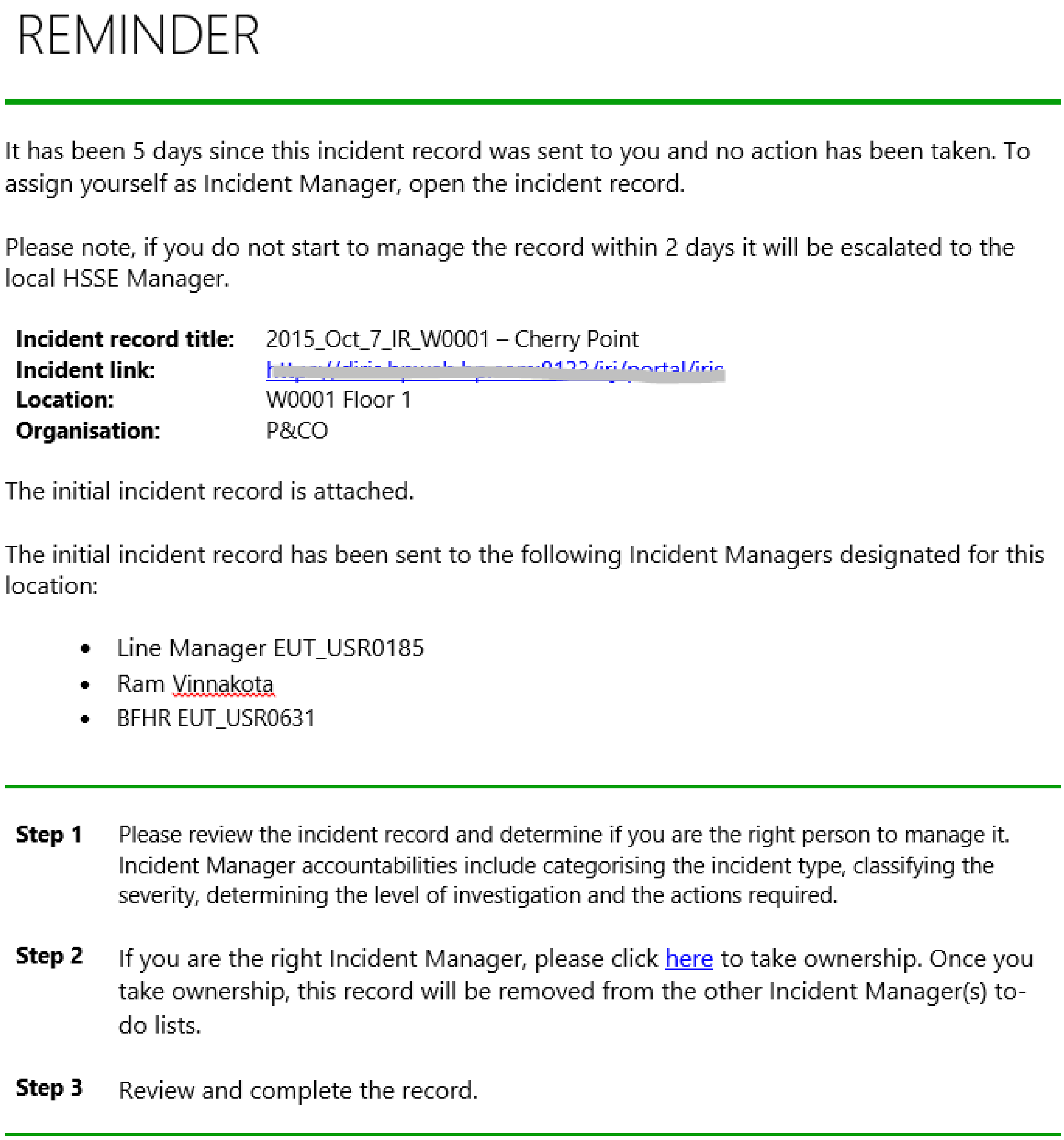
- Don't use 1x1 pixel images to control spacing as this may cause the e-mail to be classified as spam.

Chapter 5: Design & Development Methodology

I recommend the following process for the e-mail template design and development:

1. Mock up the email in Word

The easiest tool to use to capture the business requirement is Microsoft Word, and so use this product to mock-up the e-mail for sign-off. It should be understood that the final result will not be exactly the same as the mock up on all e-mail clients and devices.



REMINDER

It has been 5 days since this incident record was sent to you and no action has been taken. To assign yourself as Incident Manager, open the incident record.

Please note, if you do not start to manage the record within 2 days it will be escalated to the local HSSE Manager.

Incident record title:	2015_Oct_7_IR_W0001 – Cherry Point
Incident link:	[redacted]
Location:	W0001 Floor 1
Organisation:	P&CO

The initial incident record is attached.

The initial incident record has been sent to the following Incident Managers designated for this location:

- Line Manager EUT_USR0185
- Ram Vinnakota
- BFHR EUT_USR0631

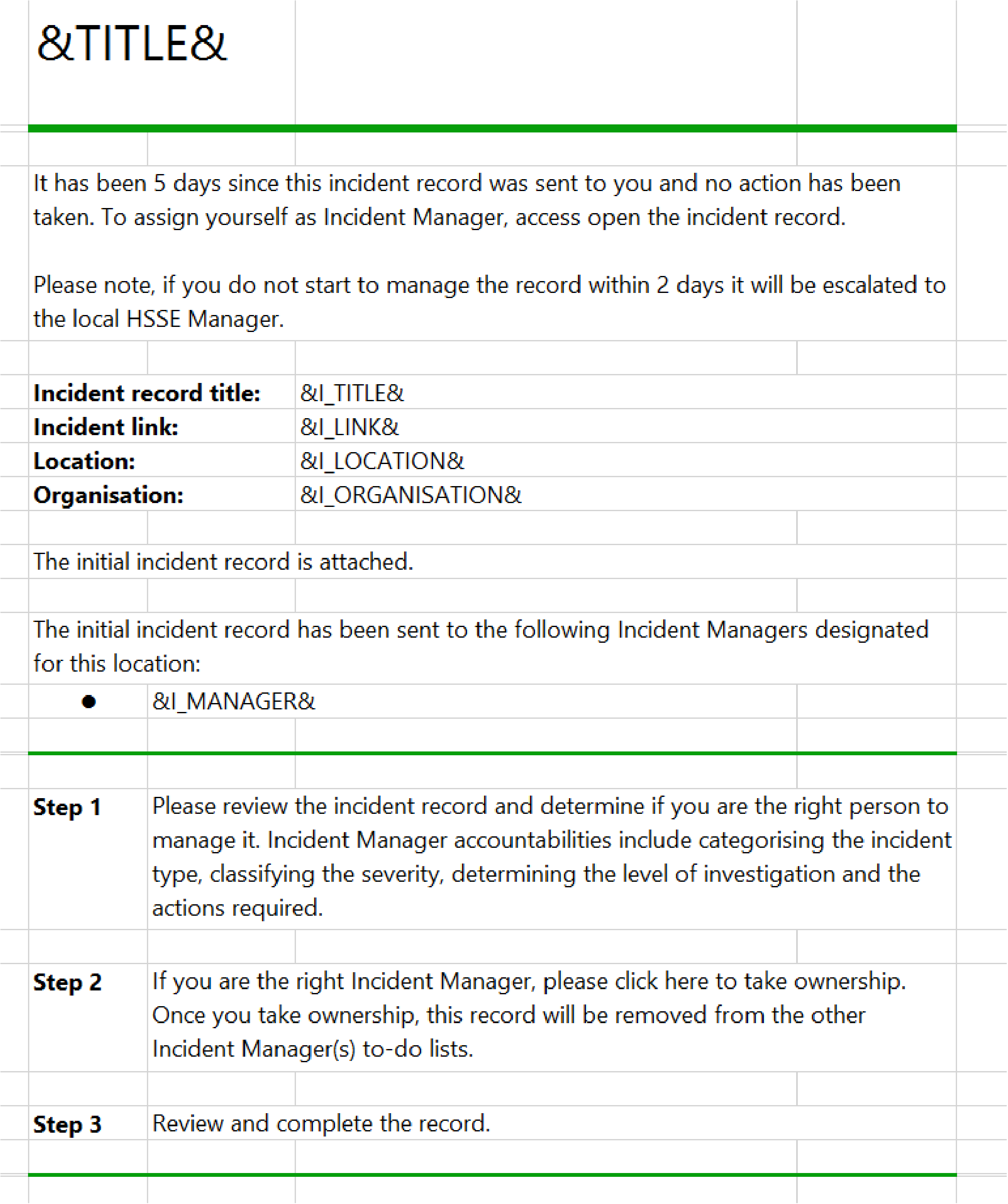
Step 1 Please review the incident record and determine if you are the right person to manage it. Incident Manager accountabilities include categorising the incident type, classifying the severity, determining the level of investigation and the actions required.

Step 2 If you are the right Incident Manager, please click here to take ownership. Once you take ownership, this record will be removed from the other Incident Manager(s) to-do lists.

Step 3 Review and complete the record.

2. Design the email in rows and columns

Since control over spacing for HTML e-mails is achieved using tables, then the entire e-mail should be designed in a tabular format. This then shows where content should be spread across multiple columns, and where template fragments should be defined. I recommend using Excel for this stage of the design, which can be given to a developer.



&TITLE&

It has been 5 days since this incident record was sent to you and no action has been taken. To assign yourself as Incident Manager, access open the incident record.

Please note, if you do not start to manage the record within 2 days it will be escalated to the local HSSE Manager.

Incident record title:	&I_TITLE&
Incident link:	&I_LINK&
Location:	&I_LOCATION&
Organisation:	&I_ORGANISATION&

The initial incident record is attached.

The initial incident record has been sent to the following Incident Managers designated for this location:

- &I_MANAGER&

Step 1 Please review the incident record and determine if you are the right person to manage it. Incident Manager accountabilities include categorising the incident type, classifying the severity, determining the level of investigation and the actions required.

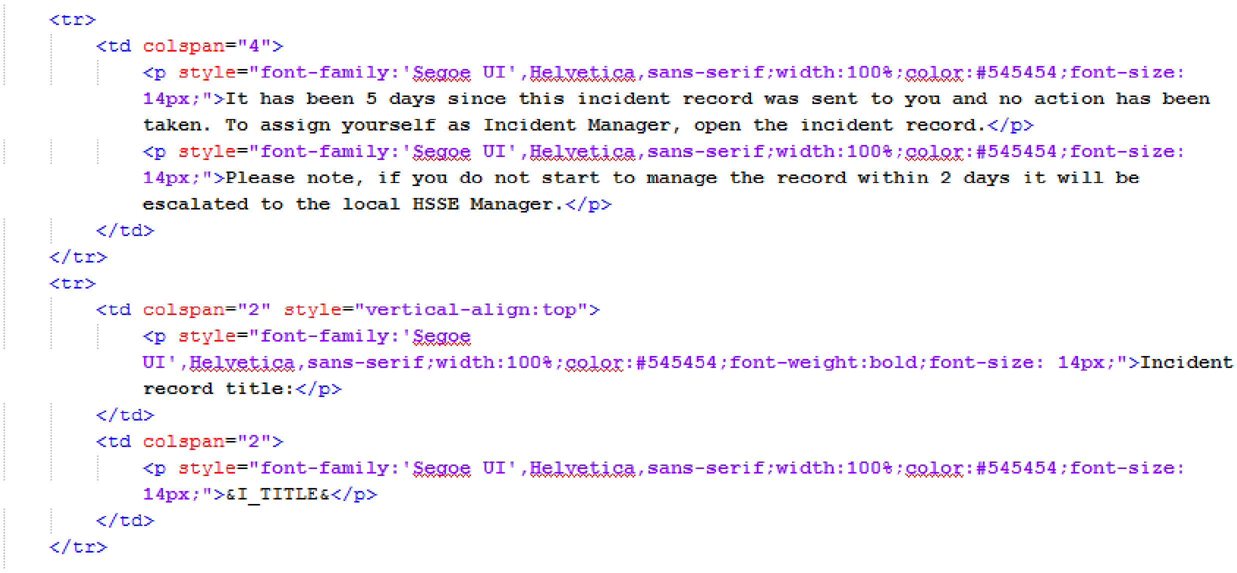
Step 2 If you are the right Incident Manager, please click here to take ownership. Once you take ownership, this record will be removed from the other Incident Manager(s) to-do lists.

Step 3 Review and complete the record.

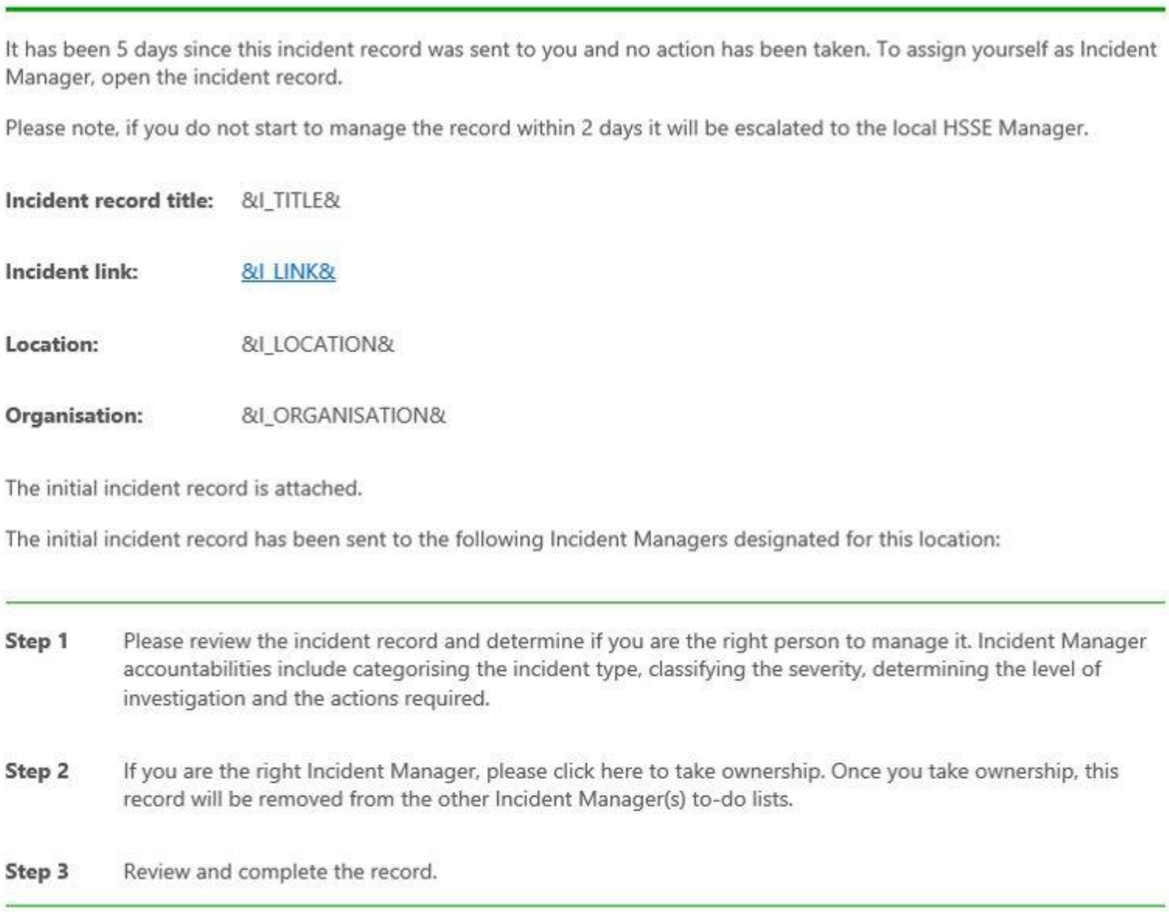
The first technical design task is to assign variable names for the dynamic text content, denoted by &…& in Floe. Discrete HTML blocks can also be assigned names, denoted by #…# in Floe.

3. Build the basic HTML template

The HTML template can be built using nested tables quite quickly based on your Excel design. Add inline styling as described in the previous chapter.



```
<tr>
    <td colspan="4">
        <p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size:
        14px;">It has been 5 days since this incident record was sent to you and no action has been
        taken. To assign yourself as Incident Manager, open the incident record.</p>
        <p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size:
        14px;">Please note, if you do not start to manage the record within 2 days it will be
        escalated to the local HSSE Manager.</p>
    </td>
</tr>
<tr>
    <td colspan="2" style="vertical-align:top">
        <p style="font-family:'Segoe
        UI',Helvetica,sans-serif;width:100%;color:#545454;font-weight:bold;font-size: 14px;">Incident
        record title:</p>
    </td>
    <td colspan="2">
        <p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size:
        14px;">&I_TITLE&</p>
    </td>
</tr>
```



It has been 5 days since this incident record was sent to you and no action has been taken. To assign yourself as Incident Manager, open the incident record.

Please note, if you do not start to manage the record within 2 days it will be escalated to the local HSSE Manager.

Incident record title:	&I_TITLE&
Incident link:	&I_LINK&
Location:	&I_LOCATION&
Organisation:	&I_ORGANISATION&

The initial incident record is attached.

The initial incident record has been sent to the following Incident Managers designated for this location:

Step 1	Please review the incident record and determine if you are the right person to manage it. Incident Manager accountabilities include categorising the incident type, classifying the severity, determining the level of investigation and the actions required.
Step 2	If you are the right Incident Manager, please click here to take ownership. Once you take ownership, this record will be removed from the other Incident Manager(s) to-do lists.
Step 3	Review and complete the record.

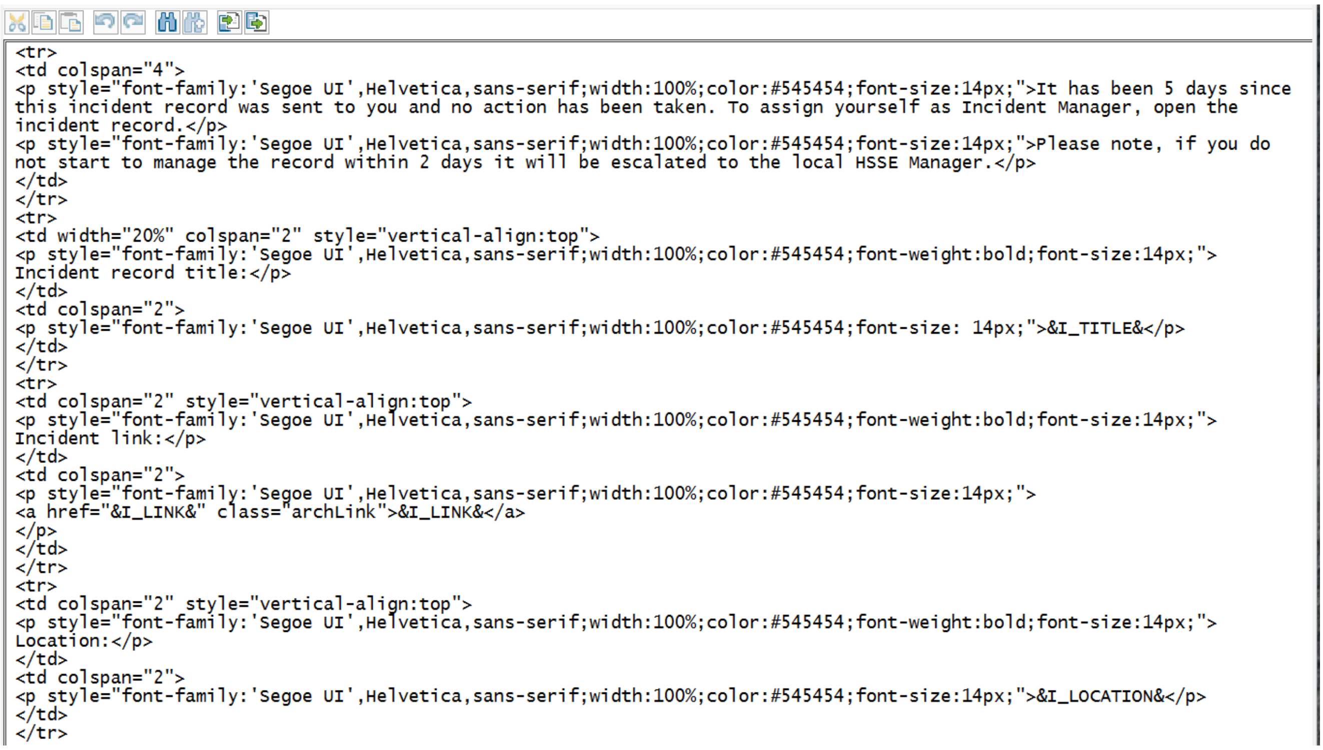
4. Convert the HTML template to fragments, and add to Floe

When a single fragment is used for the entire e-mail template, simply upload it into a single HTML block in Floe.

Where multiple fragments are required, copy and paste the fragments from the HTML editor into multiple HTML blocks.

For complex templates it is better to format the text to remove indentation and to minimise wrapping within tags.

Once the template fragments are set up, use the Floe test tool to test that the e-mail renders correctly in preview mode. All further testing should now be conducted using the Floe test tool.



```
<tr>
<td colspan="4">
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size:14px;">It has been 5 days since
this incident record was sent to you and no action has been taken. To assign yourself as Incident Manager, open the
incident record.</p>
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size:14px;">Please note, if you do
not start to manage the record within 2 days it will be escalated to the local HSSE Manager.</p>
</td>
</tr>
<tr>
<td width="20%" colspan="2" style="vertical-align:top">
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-weight:bold;font-size:14px;">
Incident record title:</p>
</td>
<td colspan="2">
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size: 14px;">&I_TITLE&</p>
</td>
</tr>
<tr>
<td colspan="2" style="vertical-align:top">
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-weight:bold;font-size:14px;">
Incident link:</p>
</td>
<td colspan="2">
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size:14px;">
<a href="&I_LINK&" class="archLink">&I_LINK&</a>
</p>
</td>
</tr>
<tr>
<td colspan="2" style="vertical-align:top">
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-weight:bold;font-size:14px;">
Location:</p>
</td>
<td colspan="2">
<p style="font-family:'Segoe UI',Helvetica,sans-serif;width:100%;color:#545454;font-size:14px;">&I_LOCATION&</p>
</td>
</tr>
```

5. Build nested fragments

Where the same piece of HTML code is repeated in a fragment / fragments, it can be split out into a separate fragment and then nested at the various points required. In some situations, entire header fragments can be common to multiple e-mail templates, and designed so that they can be re-used in entirety. In other scenarios, smaller blocks, such as logos can be managed as a separate HTML block. (See #ARCH_LOGO# in the example below.)

```
<tr>
      <td style="width:100%;">
            <p style="width: 100%; color: rgb(84, 84, 84); font-family: Segoe
UI, Verdana, Helvetica, Arial, sans-serif; font-size: 16px;">Kind
regards,</p>
            <p style="width: 100%; color: rgb(84, 84, 84); font-family: Segoe
UI, Verdana, Helvetica, Arial, sans-serif; font-size: 16px;"><strong>HR
Admin</strong></p>
            <p style="width: 100%; color: rgb(84, 84, 84); font-family: Segoe
UI, Verdana, Helvetica, Arial, sans-serif; font-size: 16px;">#ARCH_LOGO#</p>
      </td>
</tr>
```

The content of #ARCH_LOGO# is:

```
<img width="50" height="50" style="margin-bottom: 5px;" alt="Arch Logo"
 src="http://www.mywebsite/images/logo.jpg">
```

6. Conversion to variables

You can optionally set up variables for repeated styling, such as the ‘font-family’ in the example in [5.] above, and then refer to the variable instead on each line, in order to make any styling changes in a single place.

```
<tr>
      <td style="width:100%;">
            <p &PSTYLE&>Kind regards,</p>
            <p &PSTYLE&><strong>HR Admin</strong></p>
            <p &PSTYLE&>#ARCH_LOGO#</p>
      </td>
</tr>
```

The content of the variable &PSTYLE& is:

```
style="width: 100%; color: rgb(84, 84, 84); font-family: Segoe UI, Verdana,
Helvetica, Arial, sans-serif; font-size: 16px;"
```

This is a text string, not a discrete HTML block, so it is stored as a variable. However, for this particular purpose an HTML block could be used in place of a variable: Variables should be used when the content is going to be set or changed dynamically.

Here you can see that the HTML is much easier to read once the style has been changed to a variable and the logo moved to a separate fragment.

7. Add images

Where images are used within the e-mail there are two options:

i) Save the image on a public-facing web-server, and reference within the HTML image tag.

ii) Save the image on the SAP server and embed the image within the e-mail. This makes the e-mail larger, and so in general option i) is the best option.

8. Add data in user-exit or calling program

For testing purposes, the defined variables should be assigned realistic values, and a quick way to achieve this is to add some hard-coded values in the Floe data user-exit, such that the Floe testing tool will substitute the data into the e-mail template. This coding will be replaced, and ultimately the variables may be assigned in the Floe trigger program instead of the data user-exit.

The data user-exit is the place to handle any data transformations such as converting currencies to symbols, managing quantities and dates etc.

9. Add client & browser-specific logic

Now perform testing in a range of e-mail clients to check on the spacing, positioning and styling of the e-mail content. Add client-specific logic and styles for smaller screens as described in the previous chapter.

```
<style type="text/css">
      ul{   -webkit-margin-before:0em;
      }
      @media only screen and (max-width: 480px){

      .main{      width:95% !important; font-size:14px !important;
      }

      body{ margin-left:2px !important;
      }

      #tableId{   padding-right:25px !important;
      }

      }
      a.archLink{ font-family:Segoe UI;   color:#4F81BD;
      }
</style>
```

Settings for smaller screens

```
      <body bgcolor="#FFFFFF" style="width:100%;">

            <!--[if (mso)|(IE)]>
            <table class="main" style="position: relative; bottom: 30px;
width:90%; " border="0" cellspacing="0" cellpadding="10" align="center">
            <!--<![endif]-->
            <!--[if (!mso)&(!IE)]><!-->
            <table class="main" style="position: relative; bottom: 30px;
width:90%; " border="0" cellspacing="0" cellpadding="0" align="center">
            <!--<![endif]-->

                  <tbody>
```

Dynamic logic for Outlook padding

10. Add attachments in user-exit

Attachments can be added in the Floe trigger program or in the Floe user-exit. If attachments are to be added then the testing should include this, albeit as a separate integration test, after the e-mail design testing is complete.

11. Client testing and optimisation

Once the complete e-mail is displaying correctly in the preview mode, then testing should be conducted on different e-mail clients, and then any adjustments made to the HTML directly in the Floe configuration.

At the very least, testing should be conducted using Microsoft Outlook (latest version) and a small-screen device such as an iPhone. Ideally many more devices/clients should be included.

About The Author

Chris Scott has been building SAP-based solutions since 1994, firstly with Andersen Consulting and then with Arch after founding the company in 1996. Chris has extensive experience of working with organisations across a wide range of industries, bringing together his deep functional and technical knowledge of SAP's business suite and Arch's solutions. Within Arch, Chris has designed and managed many different functions including sales, marketing, accounting, operations and business solutions. His focus is building great teams, great processes and great customer solutions.

More Information

View Floe demonstration

Download the Floe Solution Brief

Visit www.floe.com for more information about Floe.

Arch

About Arch

Arch is a market-leading provider of SAP usability software. Founded in 1996, it develops software that brings together the power of SAP with the simplicity of user-friendly interfaces such as SAP Fiori®, PDF and HTML.

Arch has been recognized by Gartner as a "Cool Vendor" in the SAP ecosystem. It has partnered with SAP and Adobe, and works closely with SAP to deliver outstanding products to complement the standard SAP tools.

Gartner® cool vendor

Arch is currently supporting customers in EMEA, North America and the Middle East, and projects are supported worldwide through our partner network.

Find more information at www.arch-global.com.