



veeva CRM
Veeva CRM 18 Approved Email Content Creation
June 2013

Contents

Approved Email Content Creation Guidelines.....	4
Approved Email Templates	4
Approved Email Modular Design.....	4
Email Template.....	4
Email Fragment.....	5
Template Fragment	6
Approved Email Configuration Tokens.....	7
Embed Hyperlinks to Vault Documents.....	9
Understanding Email Properties - From Address and Email Replies	10
Understanding Email Properties - Using Email Domains.....	12
Vault Viewer Options.....	13
Understanding Product and Detail Group Properties and Content Alignment	14
Understanding Language Property.....	14
Resources:	16

Figures

Figure 1: Email Template with 2 Fragments inserted by the rep. 6

Figure 2: Relationships defined on the Email Template and Fragment document types in Vault.10

Figure 3: Define email properties in Vault Email Templates..... 11

Figure 4: Define embedded viewer landing page URL in Vault 14

Tables

Table 1: Configuration Tokens..... 9

Table 2: Token Configuration 10

Approved Email Content Creation Guidelines

APPROVED EMAIL TEMPLATES

Email Templates are selected by Reps and emailed to Accounts. Approved Email requires email templates be created and managed in Vault PromoMats. There are no Veeva-specific limitations to creating HTML email templates. However, creating email templates for Approved Email requires a knowledge of HTML and CSS, and an understanding of how to format HTML and CSS that renders correctly on the different online and offline email clients, for example hotmail.com and Microsoft Outlook.

No standard exists for creating HTML for emails as each email client supports a different subset of CSS. However, email marketing services, for example Campaign Monitor and Mailchimp are a good resource for documented best practices. They also offer free sample email templates that can be used for free and as a starting point for email designs.

The following guidelines are a starting point for creating email templates that render on all email clients:

- Use `<table>` (and nested `<table>` tag) for absolute placement. Do not rely on `<div>`.
- CSS should be inline in the head using the `<style>` tag or inline in the `<body>` HTML. Do not link to external CSS style sheets.
- Only a subset of CSS is supported in many email clients (see Resources, below).
- Images should be formatted for screen resolution and be as small as possible

Because up to 40% of all email is opened on mobile devices (phones and tablets), you should consider optimizing email templates to render on these devices. This typically follows a process called responsive design where the CSS adapts to render on both desktop and mobile email clients.

See the Resources section for links to useful guides about creating email templates.

APPROVED EMAIL MODULAR DESIGN

Approved Email has a modular design philosophy that allows a Rep to select 0 or more documents to share within a single email template. It is not required that you use the modular format, but it can be convenient when creating a large content library.

To support the modular design, Approved Email uses three documents types within Vault PromoMats:

- Email Template
- Email Fragment
- Template Fragment

Email Template

This is a standard HTML formatted email with optional image files. This is a prebuilt message (and design) that the rep cannot edit, except for optional Veeva formatting described in the

Configuration Tokens section. Email Template HTML can reference images and contain hyperlinks to content managed in Vault, or stored on other sites. When hyperlinks point to content in Vault, a specific syntax is used in the `href`.

Emails often contain image references. Images must be deployed and publicly visible on the internet. Vault PromoMats will optionally deploy and host email images. If your email HTML references images using a relative path, for example ``, Vault PromoMats can load and deploy the images to a public web server so that all email recipients can view the images. This saves the effort of having to deploy images on your servers. (To learn more, read the [Vault documentation](#).)

At least one Email Template is required for Approved Email to function. See the [Vault PromoMats](#) documentation to understand how to load HTML email templates and image assets.

Email Fragment

An Email Fragment is a small snippet of HTML that contains a specific message and optional hyperlinks to content stored in Vault, or on other websites. Fragment HTML is designed to be embedded within an Email Template; it is not sent by a Rep on its own. Fragments are, in essence, children of Email Templates.

The advantage of using Fragments is they represent a small piece of content that a Rep can select from a list and insert into an Email Template. Adding more content does not require creating more Email Templates. Create new Email Fragments to share new or updated content independently of Email Templates.

When using Fragments, the Email Template HTML must include a special Veeva token `{{insertEmailFragments}}` to enable the Add Documents button on the Send Email page online or in iRep. The Add Documents button displays a list of Email Fragments aligned to the Detail Product (and Detail Group, when used).

In order for Approved Email to insert Email Fragment HTML into an Email Template, the Fragment HTML design must match the design of the Email Template. Fragments cannot be designed independently of the template. In general, it is best to design an Email Template with the Fragment embedded and then remove the Fragment HTML from the template and save it as another document.

The following formatting guidelines apply to Email Fragment HTML design:

- CSS can inherit from the Email Template, but it is best if styles are embedded directly in the Fragment so a preview can be generated in Vault PromoMats
- Fragment HTML must be enclosed in a `<tr><tr>` tag. An Email Template must contain an empty `<table> </table>`, and within the table is the Veeva token `{{insertEmailFragments}}`. This way, Approved Email will always insert Fragment HTML as `<tr>` rows within the empty table. Fragments always stack vertically within the Template.

- Fragments can contain references to images. A separate image asset zip archive must be loaded into Vault along with the HTML for Vault to deploy the images and update the Fragment HTML to reference the absolute paths of the deployed images.

Example

- Relative image path in source HTML: ``
 - Absolute image path in Veeva HTML: ``
- Fragments can contain hyperlinks to Vault documents, or to content on other sites. When linking to Vault documents, special tokens are required in the `<href>` that leverage Relationships defined on the Vault document types. Approved Email will automatically generate the hyperlink when the email is sent. For example, `` will generate a hyperlink to the Promotional Piece identified in Vault as having a relationship to a specific Email Fragment.

For more information about Relationships, see the Vault documentation.

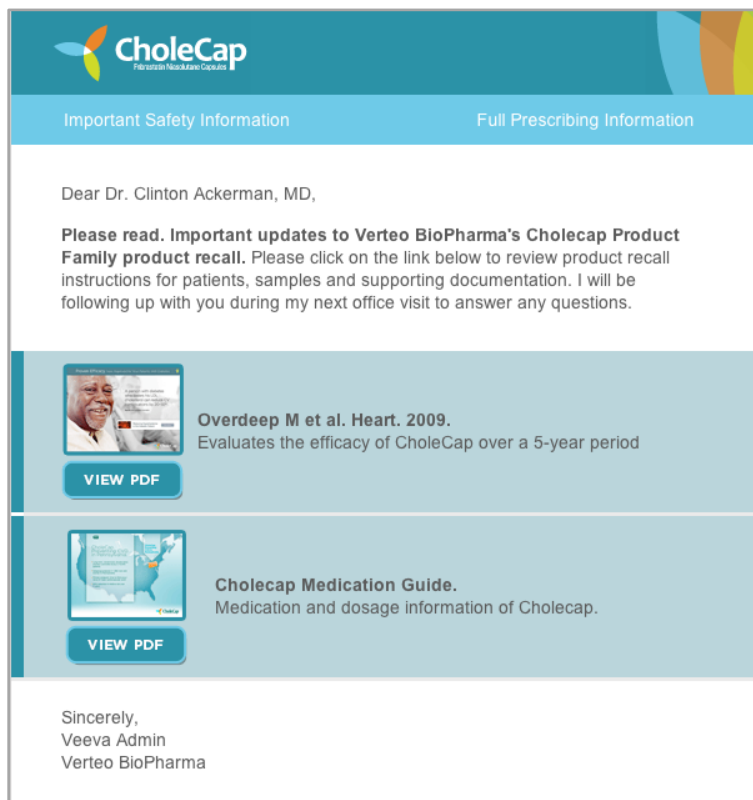


Figure 1: Email Template with 2 Fragments inserted by the rep.

Template Fragment

A Template Fragment is also a modular snippet of HTML. It is typically used to create a footer that is appended to all Email Templates. For example, you may have a standard disclaimer or ISI statement that must be included at the bottom of every Email Template. This allows you to define the disclaimer once and Approved Email will add it to all outgoing emails.

The Email Template HTML must include a special Veeva token `{{emailTemplateFragment}}` to indicate where to insert the Template Fragment HTML. This HTML should be enclosed within a nested `<table>` that fits within the overall design of the Email Template. Within Vault, you can define a specific Template Fragment to include within each Email Template.

Template Fragments are optional.

Approved Email Configuration Tokens

Approved Email tokens are used for personalization and for limited personalization by Reps. Personalization values are pulled from the Account and User objects, and allow emails to contain personalized content. Configuration tokens are used to enable specific behavior prior to an email being sent, for example:

- Reps can select from a picklist of values before sending the email, for example Sincerely or Regards
- Reps can insert free text into a section of the email. The free text is validated against an optional list of restricted words and phrases defined by the customer in the Veeva Message APPROVED_EMAIL_RESTRICTED_WORDS.

<i>Token</i>	<i>Description</i>
<code>{{accTitle}}</code>	Inserts Account salutation
<code>{{accFname}}</code>	Inserts Account first name
<code>{{accLname}}</code>	Inserts Account last name
<code>{{accCredentials}}</code>	Inserts Account credentials
<code>{{userName}}</code>	Inserts the Rep's name, typically into the 'from' or 'reply-to' Vault 'Email Template' properties of an outbound email template, or into the body of the email template for personalization.
<code>{{userEmailAddress}}</code>	Inserts the Rep's email address, typically into the 'from' or 'reply-to' fields of an outbound Email Template.
<code>{{ObjectAPIName.FieldAPIName}}</code>	<p>Expands the markup syntax when the simple tokens are insufficient. Most field types from the Account and User objects are available, but restrictions exist as outlined below. The following are examples from the User and Account objects:</p> <ul style="list-style-type: none">• <code>{{User.Email}}</code>• <code>{{Account.Specialty_1_vod__c}}</code>• <code>{{Account.Formatted_Name_vod__c}}</code>

	<p>Approved Email officially supports the following field types in markup tokens:</p> <ul style="list-style-type: none"> • Formula* • Roll-Up Summary* (Account only; formulas not available on User) • Checkbox • Currency • Date • Date/Time • Email • Number • Percent • Phone • Picklist • Text • Text Area <p>*Certain restrictions apply. See iRep guide for more details.</p>
{{customText[options]}}	<p>When present in an Email Template, allows a rep to select from a list of options in a picklist. For an administrator, this allows fewer email templates to be created, and promotes re-use while allowing a rep to configure the template to match the location or nature of the call. Examples:</p> <ul style="list-style-type: none"> • Select from a list of pre-defined greetings, for example, {{customText[good to see you again sorry I missed you]}} • Select from a list of physical settings, for example, {{customText[your practice pharmacy clinic hospital]}}
{{customText(inputlengthInt)}}	<p>When present in an Email Template, allows a rep to insert free text. Note the free text will be evaluated against an optional list of restricted terms/phrases:</p> <ul style="list-style-type: none"> • Free text with an input length limit, for example, {{customText(255)}}. This will render an HTML text input area that can accept up to 255 characters
{{insertEmailFragments}}	<p>Defines the section in an Email Template where Email Fragments are inserted. When the Rep selects documents to share, the corresponding Email Fragments for each document will be inserted and stacked vertically. This token must be inserted in an empty <table> </table> tag in the Email Template.</p>

	It is important the HTML + CSS design of the Email Fragment match the design of the Email Template so the final email renders properly.
{{emailTemplateFragment}}	This token inserts an HTML snippet that is managed by the Vault 'Email Template Fragment' document type. It's typically used to append a standard disclaimer or ISI into the bottom of the Email Template. The only constraint is the content type must be HTML and it must fit the overall style of the Email Template.
{{unsubscribe_product_link}}	<p>This token will insert an unsubscribe link in the email. The link can be wrapped in an HREF and appropriate visible text. The unsubscribe link will inherit the Product or Detail Group + Product that's associated to the Email Template. When the recipient clicks the unsubscribe link, the opt-out activity is recorded for the Account, email address, and Product or Detail Group + Product in the Multichannel_Consent_vod object. All future emails to the Account plus the email address are suppressed for this Product or Detail Group + Product until an opt-in activity is collected or the opt-out expiration date is set.</p> <p>Example:</p> <pre>Unsubscribe from all future emails about Cholecap.</pre>
{{productdetail}}	<p>Used within the Restricted Words validation feature when the {{customText(inputlengthInt)}} token is included in an Email Template. This token expands to match against the CRM Product Catalog names for detail products. For example, if 'Cholecap' is a Detail Product type, the Rep will not be able to insert the word 'Cholecap' into the free text field.</p> <p>This token can only be used within the Veeva Message APPROVED_EMAIL_RESTRICTED_WORDS. It will not be evaluated in any other way.</p>

Table 1: Configuration Tokens

Embed Hyperlinks to Vault Documents

Links to Vault PromoMats documents can be embedded in Email Templates or Email Fragments. Approved Email document tokens are used in HTML `<href>` tags that expand to authenticated hyperlinks when the email is sent. When using these tokens, Approved Email will validate that Vault documents have a workflow state that is a Steady State (this assumes the documents are approved for distribution). When true, the email is sent. When any one of them is false, Approved Email suppresses the email if any piece of content, including the Email Template, is no longer in an approved state. This ensures Approved Email only sends content that is compliant and approved.

The document tokens are included within the HTML source files for Email Templates or Email Fragments. To complement the tokens, specific relationships are required on the Email Template or Email Fragment document types in Vault. The relationships define the specific documents that should be validated and shared. The tokens define where in the HTML source the links are created, and the type of relationship to evaluate.

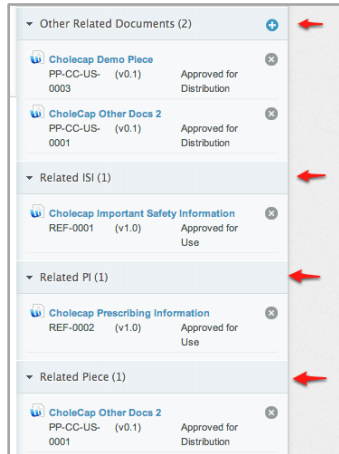


Figure 2: Relationships defined on the Email Template and Fragment document types in Vault.

Token	Description
{{PieceLink}}	Expands to a URL pointing to a Vault document of type Promotional Piece
{{ISILink}}	Expands to a URL pointing to a Vault document of type Important Safety Information
{{PILink}}	Expands to a URL pointing to a Vault document of type Prescribing Information
{{\${VaultDocID}}}	<p>Allows the content admin to include any Vault PromoMats document, however it requires a special syntax. Expands to a URL pointing to a Vault PromoMats document of any type. This requires the Vault document ID as an argument. For example <code>{{\${20}}}</code> will point to Vault document ID 20.</p> <p>The document number is visible in the URL when viewing a Vault document, and is not visible from the General Properties page sidebar. The integer to the right of #doc_info is always the unique Vault document ID. Example:</p> <p>https://customer.veevavault.com/ui/#doc_info/20/3/0</p>

Table 2: Token Configuration

Understanding Email Properties - From Address and Email Replies

Email header properties are defined in the Vault Email Template document type and pulled into Veeva CRM. For each Email Template, you may configure a unique from, (optionally) reply-to, subject, and (optionally) sending domain headers. You may want specific emails to be from the Rep (User.Email), and others from a fixed email address, for example always from brand@customer.com.

From Address

From Address is required attribute in Vault for any Email Template. This attribute can either be set to an explicit email address, or by using the {{userEmailAddress}} markup token. This attribute can be set dynamically to the sending user's email address as it exists in Salesforce.com.

From Name

This is an optional attribute in Vault for Email Templates. This attribute can be set explicitly to a static text value, or using the {{userName}} markup token can be dynamically set according to the user who is sending the email.

Reply to Address

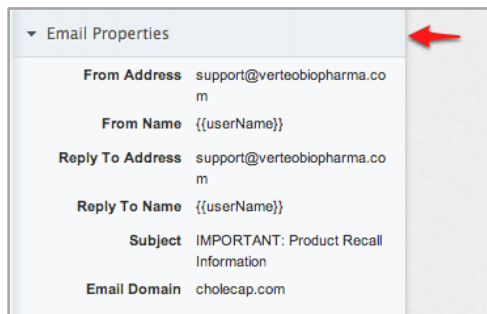
This is an optional attribute in Vault for Email Templates. This attribute can be set to an explicit email address used for all replies from the originating email. This can also be setup as an email address corresponding to a Salesforce.com inbound email handler. When this is done, the replies can be captured and processed on the Salesforce.com platform in many ways.

Reply to Name

This is an optional attribute in Vault for Email Templates. This attribute can be set explicitly to a static text value, or using the {{userName}} markup token can be dynamically set according to the user who is sending the email.

Subject

This is the subject that displays in all email clients. It is set in Vault on the Email Template and is required.



Email Properties	
From Address	support@verteobiopharma.com
From Name	{{userName}}
Reply To Address	support@verteobiopharma.com
Reply To Name	{{userName}}
Subject	IMPORTANT: Product Recall Information
Email Domain	cholecap.com

Figure 3: Define email properties in Vault Email Templates

The following are examples of how the From and Reply-to headers can be configured on Email Templates in Vault. For reference, User.Email uses the email address from the User (Rep's) record, User.Name uses the Rep's name. The examples assume the user's name is Sarah Jones and her email address is sjones@company.com.

Example 1

From displays as: Sarah Jones <sjones@company.com>

Replies go to: Sarah Jones <sjones@company.com>

	<i>From Address</i>	<i>From Name</i>	<i>Reply-to Address</i>	<i>Reply-to Name</i>
<i>Example Values</i>	User.Email	User.Name	n/a	n/a

Example 2

From displays as: Sarah Jones <sjones@company.com>

Replies go to: Customer Service <replies@brand.com>

	<i>From Address</i>	<i>From Name</i>	<i>Reply-to Address</i>	<i>Reply-to Name</i>
<i>Example Values</i>	User.Email	User.Name	replies@brand.com	Customer Service

Example 3

From displays as: Sarah Jones <name@company.com>

Replies go to: Sarah Jones <name@company.com>

	<i>From Address</i>	<i>From Name</i>	<i>Reply-to Address</i>	<i>Reply-to Name</i>
<i>Example Values</i>	name@company.com	User.Name	n/a	n/a

Example 4

From displays as: Company Mailbox <name@company.com>

Replies go to: Sarah Jones inbox@salesforce.com

Note: The following example requires a force.com inbox be setup in your org. It requires custom Apex code to manage and route the inbound email replies. It allows customers to manages replies within salesforce and maintain a conversation thread.

	<i>From Address</i>	<i>From Name</i>	<i>Reply-to Address</i>	<i>Reply-to Name</i>
<i>Example Values</i>	name@company.com	Company Mailbox	inbox@salesforce.com	User.Name

Understanding Email Properties - Using Email Domains

Approved Email allows the use of customer defined domain names when sending emails. This feature enables customers to create branded domains on behalf of which the enterprise email engine will send email. This domain will display in the <return-path> header of the email message and is validated by all receiving email servers.

There are two fundamental processes which need to be completed in order for email domains to be correctly setup and authenticated to that validations pass:

- The sending domains must be setup in the Approved Email Enterprise Email Engine
- The customer must create DNS entries that authorize Veeva to send email from these domains

- When the above steps are complete, the email domain names are available to be used in Approved Email in one of two ways:
 - Sending Email Domain field on Approved_Document_vod. Set in Vault on the Email Template, optional. When set in Vault, this value overrides the value set in the Approved Email Setting Approved Email Domain (APPROVED_EMAIL_DOMAIN_vod). This allows a product branded email template to be sent from a brand.com authenticated domain. When NULL, will default to (see below).
 - Approved Email Setting Approved Email Domain (APPROVED_EMAIL_DOMAIN_vod). This is the default when a value is not set on the Email Template. A unique domain can optionally be defined for each security Profile. The common example is for multi-country orgs where the sending domain is different for each country.

See the Approved Email Administration Guide for more details on the Email Domain configuration steps.

Vault Viewer Options

Customers have two options when deciding how Vault content displays when the Account clicks through from an email link. (When Accounts click through from links that are hosted by the customer, the experience is controlled by the customer.)

- Standalone Vault Viewer – this is the default document viewer that email recipients will use to view Vault documents. No configuration/customization is required to utilize this viewer.
- Embedded Viewer – Customers have the ability to embed the Vault document viewer in a custom portal site. This allows the customer to retain control of branding and augment the document with additional portal content. The embedded viewer is built using a small HTML snippet that will render the document within its own iFrame.

To use the Embedded Viewer in a customer portal page, the Embedded Viewer HTML Snippet (example code below) must be referenced inside the HTML of the customer's page. This can be done on one or many pages. For example, a single portal page can be used as the host for all Vault documents. Another example is one portal page per brand.com site can be configured.

To target the embedded viewer in an email, the Embedded Viewer property must be populated with a valid URL on the Vault Email Template or Email Fragment document. This attribute will be pulled into Veeva CRM and stored in the Document_Host_URL_vod field on the Approved_Document_vod object. When Approved Email processes a new email, the document links will be created such that the portal site (Document_Host_URL_vod) will be used as the landing site and ultimately the location for viewing the document.

The embedded viewer creates an iFrame. The size can be defined in pixels using the height and width parameters. An optional error message can be defined inline using the error parameter. This displays when the target is no longer in an approved or steady state in Vault.

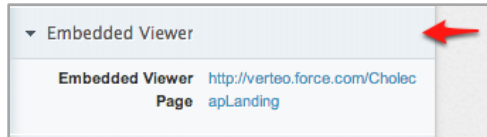


Figure 4: Define embedded viewer landing page URL in Vault

Embedded Viewer HTML Snippet

```
<div align="center" id="viewer"> </div>
<!-- The iFrame and Vault Viewer will replace the above <div> tag. Please note that the id of the <div> matches the
first parameter of the viewer constructor -->
<script>
    var tag = document.createElement('script');

    tag.src = "//js.veevavault.com/ps/vault-v1.js";
    var firstScriptTag = document.getElementsByTagName('script')[0];
    firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);

    // This function creates an <iframe> (and vault viewer)
    // after the API code downloads.
    var viewer;
    function onVeevaVaultIframeAPIReady() {
        viewer = new VV.Viewer('viewer', {
            height: '630',
            width: '712',
            error: 'We\'re sorry. The document you are trying to view is not available..'
        });
    }
</script>
```

Understanding Product and Detail Group Properties and Content Alignment

When loading new Approved Email content into Vault, there is a mandatory Product attribute that must be set for the document. Optionally, if enabled, a Detail Group can also be set for the Vault document.

These Product and Detail Group attributes are used by the Approved Email solution to accurately align Vault content to CRM end users. The Product value corresponds to the Detail Product in Veeva CRM and the Detail Group corresponds to the Detail Group in Veeva CRM. If an end user is aligned to a specific Product and/or Detail Group (brand in some cases), then by default, the end user will gain access to the approved Vault content aligned to that product.

Note: There can only be one Product and one Detail Group assigned to a single document. (Vault allows more than one Product to be assigned to a document, but this is not supported by Approved Email) CRM uses an external ID matching scheme to accurately assign the Product/Detail Group references in Veeva CRM, and the external IDs must be setup in advance in Vault and in CRM.

Understanding Language Property

The language property in Vault indicates the language in which the Vault document is written. This language value will be passed to Veeva CRM via the integration service and will be used in language filtering by the end users of the Approved Email solution. In addition to displaying the language of the document in Veeva CRM, if loaded, the preferred language will also be displayed

for each recipient during the send email process. In this case end users will be able to compare the preferred languages of on recipient to another and ultimately to the language of the document. This helps the end user understand if the best version of the content from a language perspective is being sent to the selected recipients.

HTML Examples for Templates and Fragments

Email Templates must contain a minimum structure of a standard HTML document, plus at least one `<table>` if embedding Email Fragments.

Example Email Template HTML Structure 1

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
<title>My Title</title>
</head>
<body>
content goes here. Styles are inline in the body HTML. Images are
hosted externally or by Vault.
<table>
<!-- If using Email Fragments, must have one empty table that
includes this Approved Email token -->
{{insertEmailFragments}}
</table>
</body>
</html>
```

Example Email Template Structure 2

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

<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<title>My Title 2</title>
<style type="text/css">
<!-- CSS styles are inline-->
</style>
</head>
<body>
<!-- Content goes here. Styles are inline in the body HTML.
Images are hosted externally or by Vault -->
<table>
<!-- If using Email Fragments, must have one empty table that
includes this Approved Email token -->
{{insertEmailFragments}}
</table>
</body>
</html>
```


Email Fragments must be encapsulated in a <tr></tr> tag. This is to ensure that Fragments render properly within the parent <table> tag in the Email Template. Fragments can contain images and relationships to multiple Vault documents or any number of external href links.

Note that Fragments can be shared between different Email Templates of the same product, but only if the HTML design is consistent.

Example Email Fragment Structure 1

```
<tr>
<td>
<!-- Content goes here. Styles are inline in the HTML, or can
reference styles in the Email Template <style> tag. Images are
stored externally or in Vault -->
<a href="{{PieceLink}}">This creates a link to a Vault document
type of Promotional Piece</a>
<a href="{{ISILink}}">This creates a link to a Vault document
type of Important Safety Information</a>
</td>
</tr>
```

Example Email Fragment Structure 2

```
<tr>
<td>
<table>
<tr>
<td>
<!-- Content goes here. Styles are inline in the HTML, or can
reference styles in the Email Template <style> tag. Images are
stored externally or in Vault -->
<a href="{{20}}">This creates a link to Vault document ID 20
</a>
<a href="{{PILink}}">This creates a link to a Vault document type
of Prescribing Information</a>
</td>
</tr>
</table>
</td>
</tr>
```

Resources:

- Litmus - test your email designs on all email clients, and spam filter testing
<https://litmus.com/>
- Campaign Monitor list of CSS support for all major email clients
<http://www.campaignmonitor.com/css/>
- Campaign Monitor Guidelines for HTML Formatting
<http://www.campaignmonitor.com/resources/will-it-work/guidelines/>
- Zurb guide to responsive design and examples
<http://zurb.com/playground/responsive-email-templates>
- Mailchimp guide to responsive email design
<http://mailchimp.com/resources/guides/html/email-on-mobile-devices/>