

# Liquid syntax / dynamic content

Dynamic content is exciting new functionality that enables powerful customization of marketing content. With dynamic content, you can send an email to a group of people with HTML formatted text displayed based on the recipients' tags or field values.

You may be thinking, that sounds like merge fields or decision diamonds. Well, it is like merge fields but instead of static content we can input dynamic content and it can replace the need for decision diamonds in a campaign that route people into different email sequences. Currently, this functionality is available in Infusionsoft's Broadcast Email and Campaign Builder's email templates. Also, Dynamic content is available in Keap "simple email broadcast" tool and the Keap Advanced Email tool.

For more information check out these online resources: this article at [Monkeypod Marketing](#), this article at [Shopify](#), and this one at [Github](#).

Dynamic Content is a powerful feature of our software. However, it requires knowledge of the Ruby programming language. If you are unfamiliar with Ruby, we recommend learning more about the language before trying this feature.

Keap Technical Support will not assist with programming or debugging your Dynamic Content.

## Getting started

I'm a new user, what is a simple way to immediately benefit from dynamic content?

If any of your contacts don't have a first name in the contact record, and you are using merge fields or decision diamonds to send custom communications, it can result in those contacts seeing code or something confusing in the name field. This is a poor experience for them and embarrassing for you. With Dynamic Content, you can define a default value for first name like "friend" to ensure all contacts have a similar and positive communication. Contacts that have their name in the contact record name field will see their name in custom communications, and contacts without a name in the field will see the word friend or whichever term that you choose.

## FAQ

### What is Liquid?

Liquid is a template language that enables dynamic content, somewhat similar in concept to merge fields but instead of static content we are able to input dynamic content in fields such as HTML formatted text and images.

### Is dynamic content available in Keap or Infusionsoft?

Yes, dynamic content is available in all versions of our software.

#### **Does dynamic content work with custom fields?**

Yes, it works with all fields.

#### **Where can I use dynamic content in my app?**

If you are versed in Liquid syntax, then you can create dynamic content anywhere merge fields are used in the app.

#### **What kind of dynamic content can I use?**

HTML formatted text, hyperlinks, and images.

#### **How can I learn more about Liquid and dynamic content?**

Navigate to Marketing > Dynamic Content. There is a link to a syntax guide on shopify's site. This site provides the rules that define how to apply the Liquid programming language.

#### **What can Keap Support help me with?**

At this time, our support team can help with the basics of the dynamic content editor. And if bugs are found, we will escalate to our product team.

## Liquid syntax

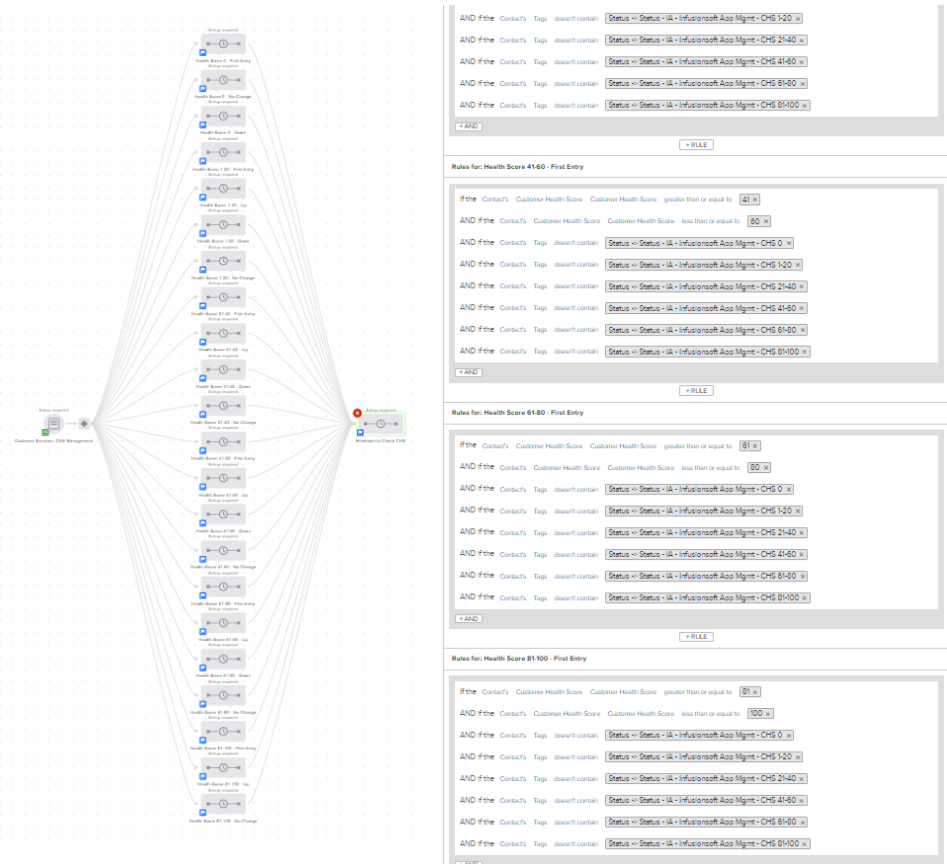
Liquid is a template language created by [Shopify](#) and written in Ruby. It is now available as an open source project on [GitHub](#), and used by many different software projects and companies. Liquid is the backbone of all Shopify themes, and is used to load dynamic content to the pages of online stores. (Source: <https://shopify.dev/docs/themes/liquid/reference>).

Designers and developers can use template language to combine static and dynamic content to populate pages with data from a Shopify store. Static elements are written in HTML and dynamic elements are written in Liquid.

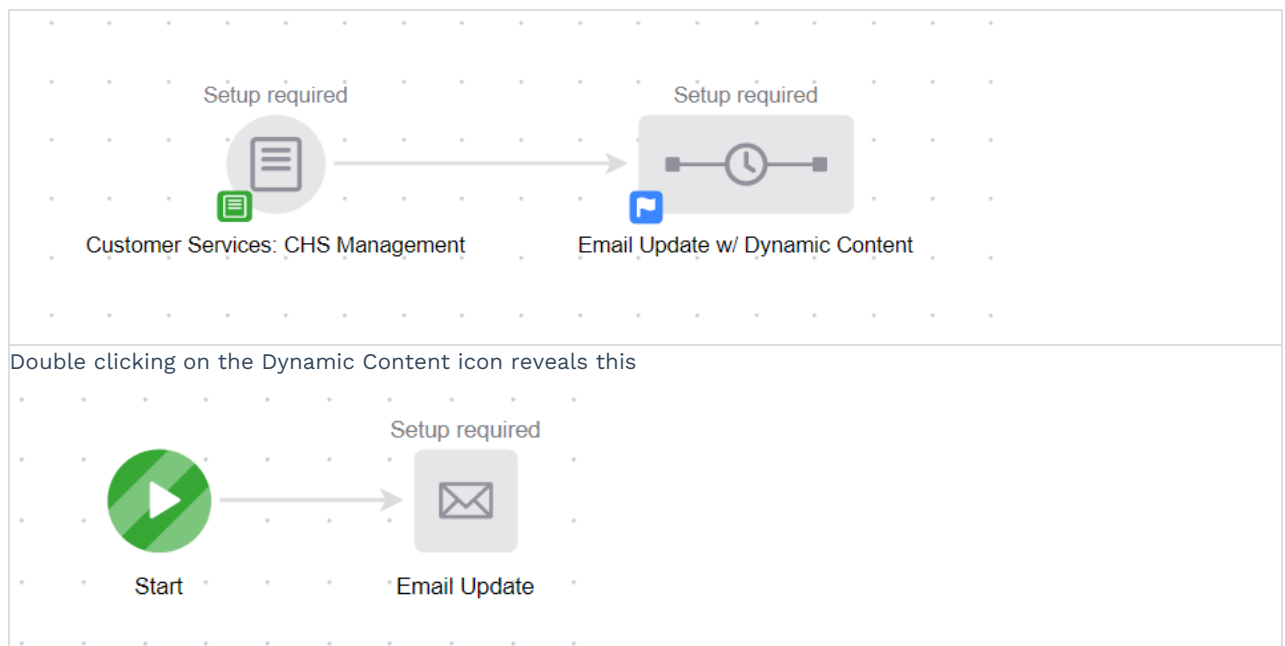
Infusionsoft's **Broadcast Email** and **Campaign Builder's email** are where Liquid syntax is currently available for use.

For more information on Liquid syntax please see the open source documentation on [Shopify's Page](#).

Prior to Liquid Content, your campaign in Campaign Builder might look something like this...



After, your campaign may look like this:



## Liquid code basics

How Keap implemented the feature:

Because some merge fields are immediate, while some are sensitive to the time which an automation might occur,

we've designed two distinct phases of how liquid will execute.

- **Phase 1** - Resolves at time of campaign publish. Looks like this `{{ merge field }}`
- **Phase 2** - Resolves at time of automation execution. Looks like this `[[ merge field]]`

The best way to explain this is with today's date.

`{{ today.date_and_time }}`

If I were to include this in an email within a campaign. And I published the campaign today. It would contain the date the campaign published.

However, if I were to use `[[ today.date_and_time]]` in the email, it would use the date and time that the email was sent. If the campaign published last week, but this email was sent yesterday, it would merge yesterday's date and time.

<p><b>Objects</b></p> <ul style="list-style-type: none"> <li>• Tell Liquid where to show content</li> <li>• Denoted by double curly braces: <code>{{ and }}</code> or <code>[[ and ]]</code></li> </ul>	<p><b>Operators</b></p> <ul style="list-style-type: none"> <li>• <code>==</code> equal to</li> <li>• <code>!=</code> not equal to</li> <li>• <code>&gt;</code> greater than</li> <li>• <code>&lt;</code> less than</li> <li>• <code>&gt;=</code> greater than or equal to</li> <li>• <code>&lt;=</code> less than or equal to</li> <li>• <code>or</code> this OR that</li> <li>• <code>and</code> must be this AND that</li> <li>• <code>contains</code> contains</li> </ul>
<p><b>Tags</b></p> <ul style="list-style-type: none"> <li>• Create the logic and control flow for templates</li> <li>• Denoted by curly braces and percent signs: <code>{% and %}</code></li> </ul>	
<p><b>Filters</b></p> <ul style="list-style-type: none"> <li>• Change the output of a Liquid object</li> <li>• Used within an output and are separated by a <code> </code></li> </ul>	

### Examples of liquid basic syntax

Code	Result
<code>{{ contact.firstname }}</code>	<i>George</i>

### Fallback values; i.e. defaults

Code	Result
<code>Hi {{ contact.firstname   default: 'there' }}</code>	<i>Hi George</i>
	If Contact does not have a first name: <i>Hi there</i>

### Control flow tags - based on number of employees

Code	Result
{% if contact.numberofemployees > 50 %}	<i>With a large organization like yours, communication is key.</i>
{% elsif contact.numberofemployees > 5 %}	<i>In SMBs, the key is balance.</i>
{% else %}	<i>In a micro-org like yours, you need to save every dollar.</i>
{% endif %}	

#### Control flow tags - based on email domain

Code	Result
{% if contact.email1.address contains '@gmail.com' %}	<i>Gmail sync for Keap captures the email communications in Gmail into your contact record.</i>
{% else %}	<i>Gmail sync and Microsoft email sync for Keap captures email communications into your contact records.</i>
{% endif %}	

#### Dates

Code	Result
{{ today.date }}	<i>2019-08-13</i>
{{ today.date   short }}	<i>8/13/19</i>
{{ today.date   medium }}	<i>Aug 13, 2019</i>
{{ today.date   long }}	<i>August 13, 2019</i>
{{ today.date   full }}	<i>Tuesday, August 13, 2019</i>
{{ today.date   plus_days: 14   full }}	<i>Tuesday, August 27, 2019</i>

#### Case statements

```

Hello ~Contact.FirstName~,

~Date.DayOfWeek~

{% capture day %}{{ 'now' | date: '%A' }}{% endcapture %}

{% case day %}

{% when 'Sunday' %} Check out our Sunday specials!

{% when 'Monday' %} Check out our Monday specials!

{% when 'Tuesday' %} Check out our Tuesday specials!

{% when 'Wednesday' %} Check out our Wednesday specials!

{% when 'Thursday' %} Check out our Thursday specials!

```

```
{% when 'Friday' %} Check out our Friday specials!
```

```
{% when 'Saturday' %} Check out our Saturday specials!
```

```
{% endcase %}
```

## Date and time fields

Date and time fields are stored in EST Timezone (GMT -04:00, Eastern Time - US & Canada. When you return these fields in the liquid editor, the date or time is being converted to match the application's timezone which makes the date or time appear off by the difference between the application's timezone and EST. This only affects the Long, Medium, Short, and Full Modifier.

## Use Liquid code in your app

You can create dynamic content anywhere merge fields are used in the app. For example, the case statements code above could be included in a broadcast like this:

**Subject Line:**  Merge

**Preview Text** ⓘ:  Merge

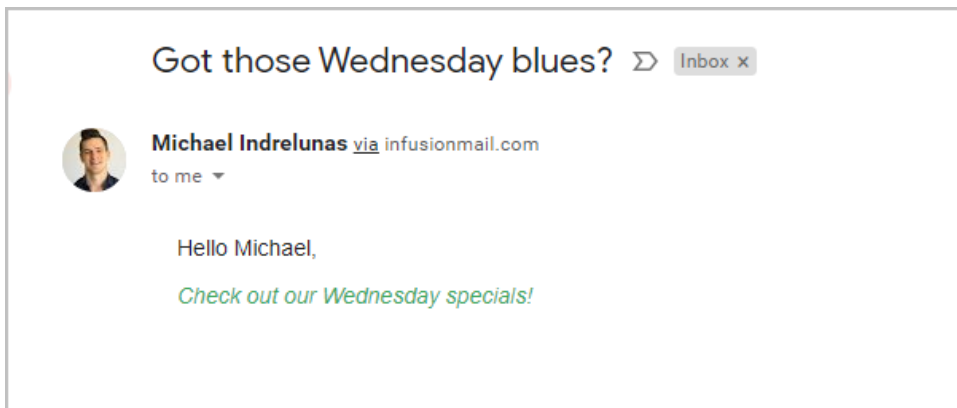
↑ DETAILS

---

Hello ~Contact.FirstName~,

```
{% capture day %}{{ 'now' | date: '%A' }}{% endcapture %}
{% case day %}
{% when 'Sunday' %} Check out our Sunday specials!
{% when 'Monday' %} Check out our Monday specials!
{% when 'Tuesday' %} Check out our Tuesday specials!
{% when 'Wednesday' %} Check out our Wednesday specials!
{% when 'Thursday' %} Check out our Thursday specials!
{% when 'Friday' %} Check out our Friday specials!
{% when 'Saturday' %} Check out our Saturday specials!
{% endcase %}
```

And your clients would receive an email like this:



## My code preview shows an error message / blank / code

If dynamic content is not working as expected, the code preview is likely showing either blank, code, or an error message instead of the expected text. While the Keap Support team does not provide support for HTML and other code, here are some high level options:

The HTML preview **shows my code instead** of the dynamic content.

- Check the code. It is likely that you have incorrectly entered liquid syntax.
- You can either read it to scan for accidental addition or deletion of text or you can start over and recreate it.

The HTML preview **shows blank instead** of the dynamic content.

- Check the code. Your liquid syntax is probably correct, but your conditions are resolving to a null state.
- Start with reviewing what your code says is supposed to happen if the conditions are true or are false. One of those may be empty and would show up as a blank result.
- If that wasn't it, look at the logic of what your code is trying to do and verify that your data supports it for both true and false scenarios.

The HTML preview **shows an error message instead** of the dynamic content.

- If you enter invalid content into a field, it may result in an error message. Walk through what type of results you are looking for and the types of fields you are using to ensure the logic.
- For example, we had a bug where if someone used a whole number custom field to do a mathematical conditional check, it would give the error: Unable to merge values.

Note! For questions about how to use this feature, please [contact our Support Team](#). To provide feedback on the accuracy of this article, use the form below.