



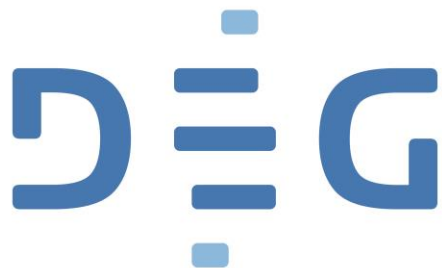
#CNX14

Flexible Content Areas with XML Interfaces

*Adam Spriggs, @wvpv
ExactTarget Engineer, DEG*



salesforce



SMARTER DIGITAL

2x ET Partner of the Year

CX14 Gold Sponsor

@DEGDigital



Questions for you.



Overview

1

Examples &
Challenges

2

Solution
Components

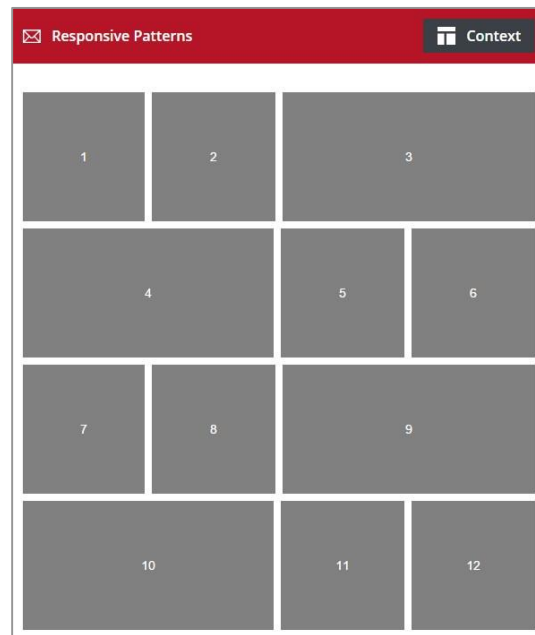
3

Show Me the
Code!



Examples

- Abandoned Cart Emails
- Weekly Sale Emails
- Dynamic, grid-based content
- ResponsiveEmailPatterns.com

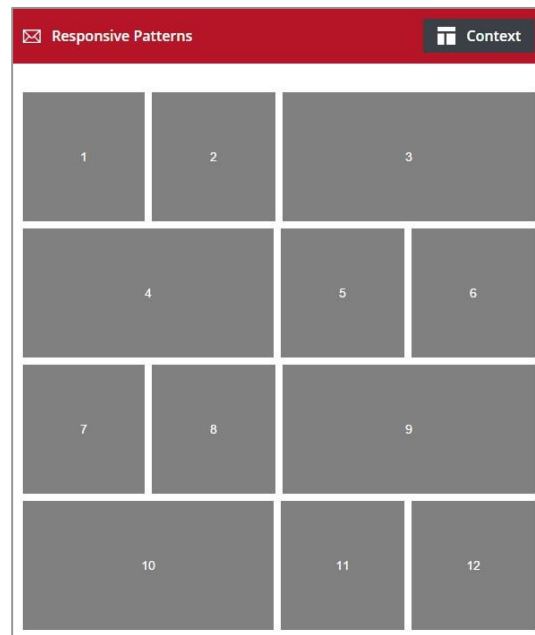


bit.ly/grid-block



Challenges

- Iterating and building HTML content
- Multiple, external data sources (e.g. abandoned cart items + upsell items)
- Sharing components across emails
- Repetition of code in an email



bit.ly/grid-block



Is there a better way to display a grid of external content in an email without repeating a significant amount of AMPScript?



A Solution

1

Retrieve your content from a data extension

3

Include a single content area multiple times, varying the parameters in each instance

2

Iterate and concatenate your data into an XML string

4

In that content area, retrieve the content to display from the XML



Benefits

1

Re-usable content area for consistency across emails

3

This method can be used to display anything, not just products

2

Like creating your own AMPScript function

4

Leveraging XML gives you some array operations



Solution Components

Data Extension



Email



Content Area





The Product Data Extension

SKU	ProductImageURL	AvgCustomerRating	ApprovedForEmail
1234	http://placeholder.it/150x150&text=1234	4.1253	true
2468	http://placeholder.it/150x150&text=2468	3.7851	true
1357	http://placeholder.it/150x150&text=1357	5.1234	true



The Email

- 1 Retrieve the result-set(s) via **LookupOrderedRows()**

```
set @products = LookupOrderedRows("ProductDE",3,  
"AvgCustomerRating desc" ,"ApprovedForEmail","true")
```



The Email

2

Iterate through the result set and build XML nodes with the **Concat()** function

```
set @productXML =  
concat(@productXML,"<productImageURL><![CDATA["  
@productImageURL,"]]></productImageURL>")
```



The Email

3

Include content area instances with the **ContentAreaByName()** function

```
%%[ set @productPos = 1 ]%%
```

```
%%=TreatAsContent(ContentAreaByName("My  
Contents\ProductDisplay"))=%%
```



The Content Area

4

Set variables with the **BuildRowSetFromXML()** function using XPath by index and name

```
set @node =  
BuildRowSetFromXML(@productXML,concat("/products/product[",  
@productPos, "]/productImageUrl"),1)
```

XPATH reference: bit.ly/et-xpath



The Content Area

- 5 Retrieve the value using **Row()** and **Field()** functions

```
set @productImageURL = Field(Row(@node, 1), 'Value')
```




The Content Area

6

Display the value

```

```



Show Me the Code!

For this Session:



My ExactTarget Posts:





Let's Review

1

Make modular components in ExactTarget that are simple to maintain

2

Lookup
Iterate & Concatenate
Include
Retrieve & Display

3

Get the code:

bit.ly/grid-block
bit.ly/et-xpath
bit.ly/ca-xml
bit.ly/wvpv-et



Questions?



CUSTOMER JOURNEY
SHOWCASE



MARKETING
THOUGHT LEADERS



EMAIL MARKETING



PRODUCT STRATEGY
& ROADMAP



SALESFORCE FOR
MARKETERS



PERSONAL
TRANSFORMATION
& GROWTH



SOCIAL MARKETING



MOBILE & WEB
MARKETING



CREATIVITY &
INNOVATION



ROUNDTABLES



DEVELOPERS



HANDS-ON
TRAINING



INDUSTRY
TRENDSETTERS



CONNECTIONS | 2014