Tip 1: Enhanced Lists

Jennifer Dungan

Optimization Manager Analytics - Torstar

Problem

We are limited to 3 proper list dimensions, or risk data truncation by using "list supported" props

```
// A list variable configured with a comma as a delimiter
s.list1 = "Example value 1,Example value 2,Example value 3";
```

There are many times where we need to track multiple values, but these values are part of the same logical group / type.

```
// List prop delimited with a comma
s.prop1 = "value1,value2,value3";
```

Current Options / Limitations

Props (List Enabled)
Pros
You can change any prop to a list with a simple change in the configuration
You can set up many "lists" so long as you have props to use (up to 75)
Cons
Limited to 100 characters *total* (risk of truncation is high)
Limited to Hit scope (unless using Workspace Attribution Modeling)
Multiple repeated values are de-duplicated

Lists
Pros
You can pass as many items as you want
Each item passed into the list can hold up to 255 bytes of data
Can use standard eVar attribution modeling (as well as Workspace Attribution Modeling)
Cons
Limited to only 3 list dimensions
Multiple repeated values are de-duplicated

Considerations

We have two options available to us, and which solution to use will depend on a few factors:

Qualifiers:

Simple Solution:
- No repeated values in your list
- No need to have specifically correlated events for each *individual* list item
- No need for correlated dimensions to each list *individual* item

Advanced Solution:
- *Require* repeated values in your list
- Need to correlate specific counters / events (including repetitions as per above) for each *individual* list item
- Need for correlated dimensions to each *individual* list item

Solution (Simple Implementation)

Use one List Dimension to store multiple types of data, using prefixes to identify uses and Classifications to split the values to each use:

Example:

- Asset Tag - **tag:{tagname}**

- Category - **cat:{categoryname}**

- Author and Author UUID - **auth:{authorname}:{authoruuid}**

- Custom Properties **cust:{property}**

- etc

** prefix for identification | value for tracking

```
s.list1 = "tag:wire,tag:imported,cat:news,cat:news|world,auth:john smith:d12e3a94-68fc-42e0-9c9d-1f0705b9e1ac,cust:sponsored-content";
```

Solution (Simple Implementation)

Classification Rule Builder Test

Rule Builder / Demo: General List / Testing: Demo: General List   ❓ Documentation

| Rules Tested: 5 | Keys Tested: 6 | Unmatched Keys: 0 |

Sample Keys   **Run Test**

🔍 Filter   Return to Rule Set

tag:wire
tag:imported
cat:news
cat:news|world
auth:john smith:d12e3a94-68fc-42e0-9c9(
cust:sponsored-content

Key	Author	Category	Tag	Custom Property	Author UUID
tag:wire			wire		
tag:imported			imported		
cat:news		news			
cat:news|world		news|world			
auth:john smith:d12e3a94-68fc-42e0-9c9d-1f0705b9e1ac	john smith				d12e3a94-68fc-42e0-9c9d-1f0705b9e1ac
cust:sponsored-content				sponsored-content	

Adobe Summit

Solution (Simple Implementation)

General List

General List
Page: 1 / 1 Rows: 10 1-6 of 6

1. tag:wire
2. tag:imported
3. cust:sponsored-content
4. auth:john smith:d12e3a94-68fc-42e0-9c9d-1f0705b9e1ac
5. cat:news
6. cat:news|world
7. author:jane doe:cc05269c-3b40-4864-ae2a-3e0ccf31ab20
8. cat:sports
9. cat:opinion
10. cat:news|world|europe

2,396
1,862
1,594
1,368

General List - Tag

Tag	Page Views
Page: 1 / 1 Rows: 5 1-2 of 2	
	↓ 14,684
1. wire	
2. imported	
3. canadian press	
4. associated press	
5. freelance	

General List - Author

	Page Views
Author	↓ 15,688
Page: 1 / 1 Rows: 5 1-5 of 5	
	↓ 3,683
1. john smith	3,683
Author UUID	3,683
Page: 1 / 1 Rows: 5 1-1 of 1	2,396
1. d12e3a94-68fc-42e0-9c9d-1f0705b9e1ac	↓ 2,396
2. jane doe	2,396
Author UUID	1,354
Page: 1 / 1 Rows: 5 1-1 of 1	
1. cc05269c-3b40-4864-ae2a-3e0ccf31ab20	↓ 1,354
3. maureen biologist	1,354
Author UUID	
Page: 1 / 1 Rows: 5 1-1 of 1	
1. 5d1cf502-5bcc-4ec0-9847-6bf6a420f9e2	

john smith
jane doe
maureen ...

Adobe Summit

Solution (Advanced Implementation)

Use Product List to pass your items along with Merchandising eVars and events.

Use the "product category" to denote the **use**, similar to how we used prefixes in the Simple Solution.

We can use the "product name" to identify our items.

Since this will never be paired with the Purchase event, we can completely ignore the next two positions in the Product List notation (quantity and price).

Now, we can use numerical Merchandising Events to support counting the specific items as many times as needed, and Merchandising eVars to pass correlated values specific to each individual item.

NOTE: This can still be used with your normal cart data, you don't have to choose one or the other.

You might need to make some adjustments to make them work together!

Solution (Advanced Implementation)

category: "wall"
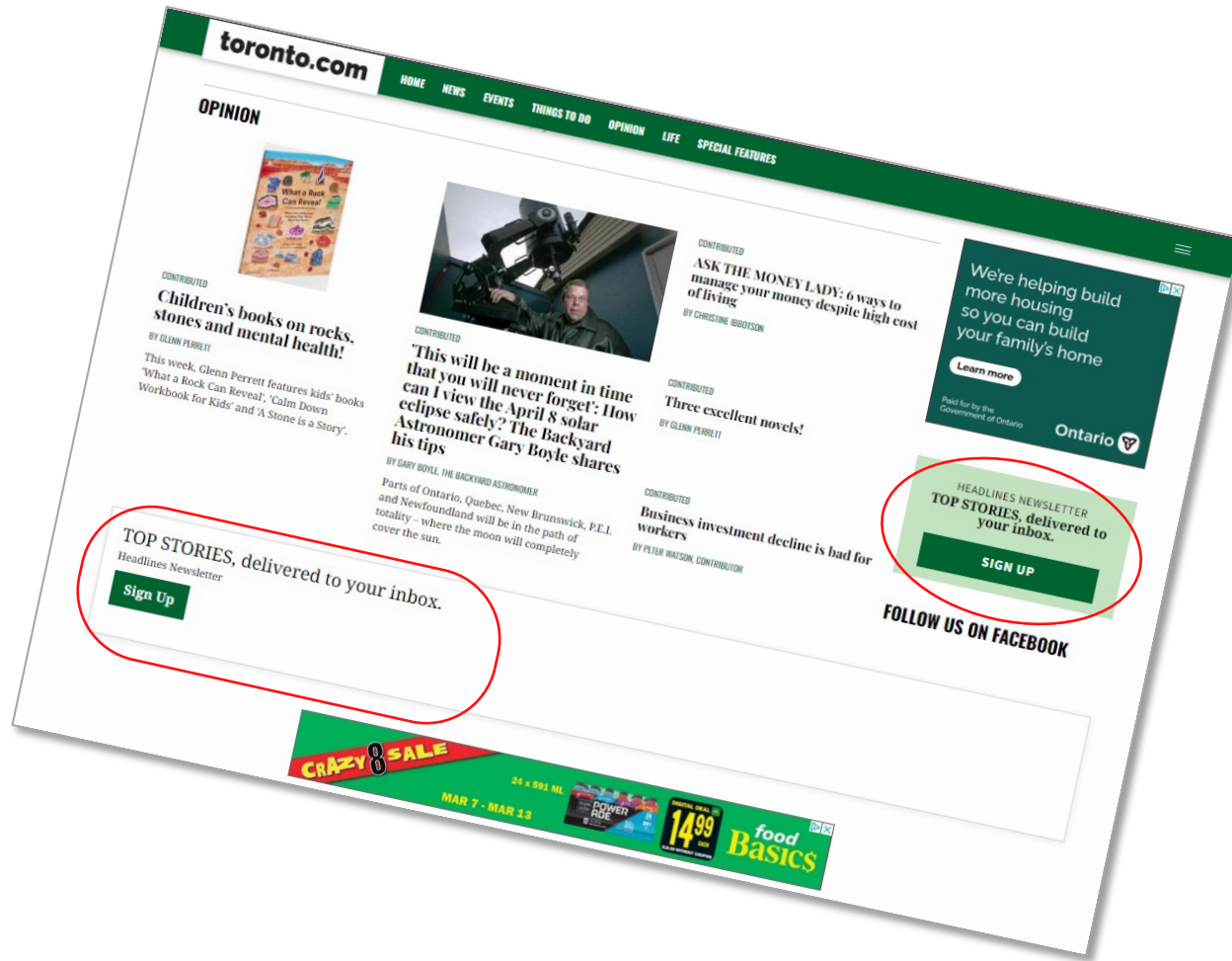product: "7d049216-aa80-4f52-abd7-78cb4f3c304f"
events: "event1=1"
merchandising eVars:

- eVar1 = "hardlock wall"
- eVar2 = "q1-2024-sale"
- eVar3 = "7d049216-aa80-4f52-abd7-78cb4f3c304f"

```
s.products = "wall;7d049216-aa80-4f52-abd7-
78cb4f3c304f;;;event1=1;eVar1=hardlock wall|eVar2=q1-2024-
sale|eVar3=7d049216-aa80-4f52-abd7-78cb4f3c304f";
```

Solution (Advanced Implementation)

category: "promo"
product: "newsletter"
events: "event2=2"

```
s.products = "promo;newsletter;;;event2=2";
```

Solution (Advanced Implementation)

Summary

1
Challenge

Current list offerings create limitations in tracking

2
Solution

Using some creative solutions such as classifications and extending products to "non-shopping cart" uses allow us more freedom

3
Result

Now we can track much more than we ever could before