

Tackling Quality Control with XSL

Jesse Clark

Webmaster & Manager of Web Communications
University of Northern Colorado (UNC)

go.unco.edu/outc18

JessClark.com

OUTC18

Web Communications @ UNC

- Web Communications is under University Relations which handles **marketing**.
- 12,000+ students, both Grad & Undergrad
- The team consists of two full-time people and two students, and support **400+ content contributors**.



Our OU Campus Instance

- Started using OU Campus in November 2015 and began converting sections in March 2016. Almost 100% converted.
- Converted from Dreamweaver Templates.
- We have **one primary (OU Campus) site**, and two sub sites for sub domains. Primary site has 190+ “sections”.

About this presentation

- Code is available here: <http://github.com/jessgusclark/outc18>
- Broken out into sections
- Questions between sections.
- This is *kinda* a technical presentation, if you are bored, dogs →



A quick warning about the code.

- Code, again: <http://github.com/jessgusclark/outc18>
- The code examples provided are just that, examples.
- **Test EVERYTHING** before deploying into production.
- Just because you can do something does not mean you should do something.
- Consider these **ideas to explore**, not solutions.

I believe that our Content Authors want to do the right thing.

- None of our 400+ content authors work on the web exclusively. They do other stuff in addition to the web.
- Majority of the 400+ content authors are level 8.
- The content authors want to update their websites and move on to other things.
- There are a lot of *things* to think about and usually quality control is not one of them.

Can we automate QC issues?

- Small team that doesn't have time to worry about small things. However, many small things can add up to big things.
- Let's explore how XSL can help fix Quality Control Issues before pages get published to the server.

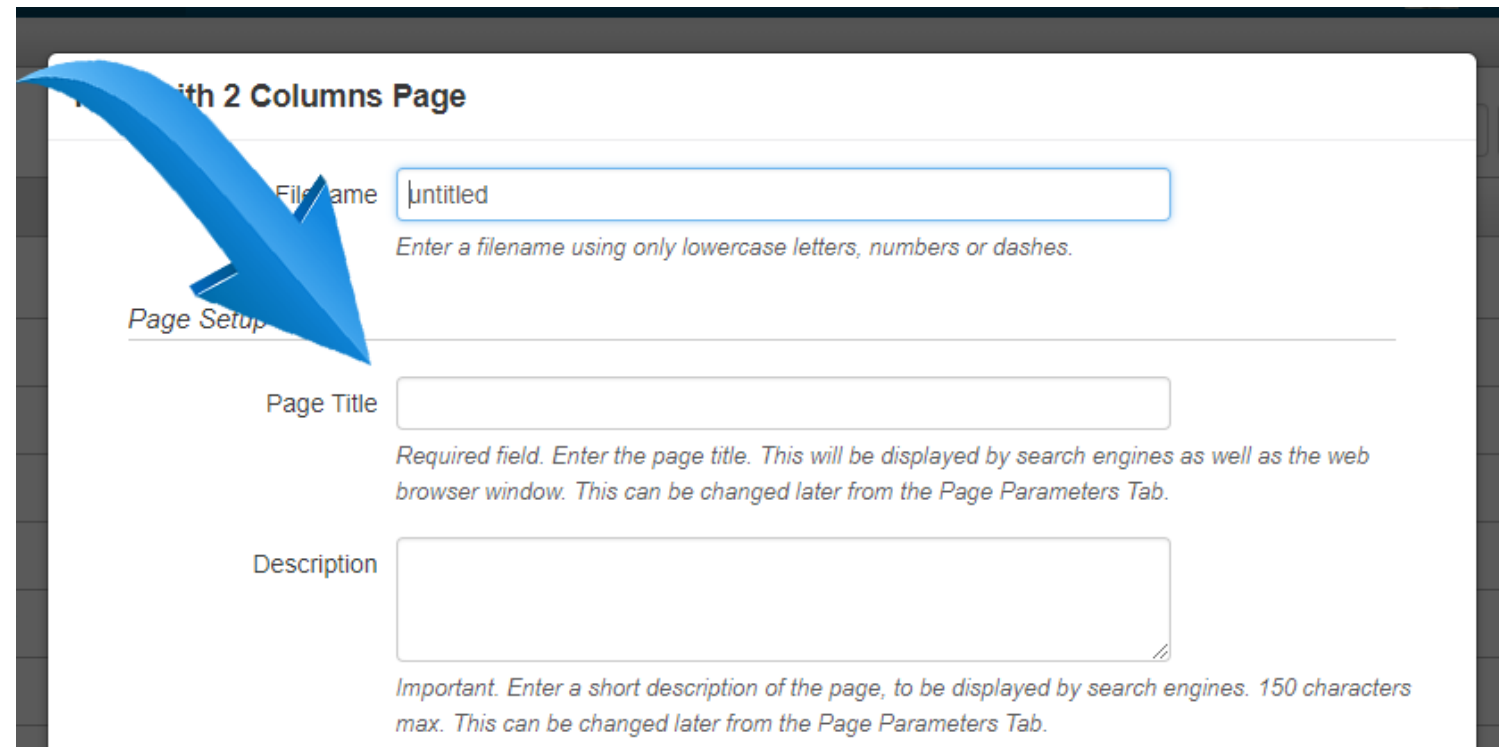
Missing <title> and <meta description> tags

This section helps users who want to do the right thing but sometimes forget because the web can be difficult.

OUTC18

There is no way someone forgot the title, right?

- The content author has to type in the title when they create the page so the <title> tag should NEVER be blank.



The screenshot shows a form titled "with 2 Columns Page". It contains three input fields: "File name" (containing "untitled"), "Page Title", and "Description". A large blue arrow points from the "File name" field down to the "Page Title" field. Below each field is a small instruction text.

File name
Enter a filename using only lowercase letters, numbers or dashes.

Page Setup


Page Title
Required field. Enter the page title. This will be displayed by search engines as well as the web browser window. This can be changed later from the Page Parameters Tab.

Description
Important. Enter a short description of the page, to be displayed by search engines. 150 characters max. This can be changed later from the Page Parameters Tab.

For 'fun', try a Find in your entire site for:

Options ?

Type Literal Text Regular Expression

Find 

Ignore Case

Replace With

Really? 42 Empty Title tags???

Find Summary			
Search Type	Literal Text	Files Searched	16787
		Files With Matches	42
		Total Matches	42
Search String			

Create a “fake” title with XSL

```
10 <title>
11   <xsl:variable name="pageTitle"
12     select="/document/ouc:properties[@label='metadata']/title">
13   <xsl:choose>
14     <xsl:when test="$pageTitle = ''">
15       <xsl:variable name="props-content"
16         select="document($path-props)/document/ouc:properties[@label='config']" />
17       <!-- create fake title based on filename and section title -->
18       <xsl:value-of select="replace(replace($ou:filename, '.aspx', ''), '-', ' ')" /> -
19       <xsl:value-of select="$props-content/parameter[@name='section-title']" />
20     </xsl:when>
21     <xsl:otherwise>
22       <!-- pull in the title the content author used -->
23       <xsl:value-of select="$pageTitle"/>
24     </xsl:otherwise>
25   </xsl:choose>
26 </title>
```

Notes about the previous code

- `$pageTitle` may already be defined in `ouvariables.xsl`
- `$props-path` was defined in `ouvariables.xsl` as well. Our `props.pcf` file includes a parameter for "Section Title" that is grabbed.
- Example: A file sitting in the Office of Admissions section called `costs.aspx` with no `<title/>` gets rendered as:
`<title>costs - Office of Admissions</title>`
- Not an ideal title, but much better than : `<title></title>`

What about auto inserting a meta description?


- I looked into text generators such as [Smmyr.com](https://smmyr.com)
- TLDR versions of pages, could this be used to create a Meta Description tag? Probably, would the generated content be better than nothing?
- Seems like a fun R&D project, but a bit more work than I wanted to do.

Perhaps it isn't a problem?

- Try this:

Options ?

Type Literal Text Regular Expression

Find 

Ignore Case

Replace With

Yeah... it was/is a problem

- *Results were too embarrassing to show.*
- *Let's call it 43% were missing.*



Solution, annoyingly remind the user that a description was missing!

The screenshot shows the UC Campus Content Management System interface. The top navigation bar includes links for Dashboard, Content, Reports, and Add-Ons. The user is logged in as Jesse Clark. The current page is 'Content > Pages > /omnidev/jesse/advanced-snippets.pcf'. The interface includes a toolbar with buttons for Preview, Edit, Source, Properties, Versions, a checkmark, a camera, a lightbulb, and a Publish button. A dropdown menu shows 'www (default)' and 'Interior Pa'. A red banner displays a 'Quality Control Check' message: 'It looks like the Description field in the 'Title and Metadata' section is missing. This field is important to help search engines find and understand what pages are about. Adding a description not only helps this page but pages that are linked to and from it. Please add a short sentence that describes the page and its purpose. [Go to the parameters of the page] [Learn more about Meta tags]'. Below the banner is the University of Northern Colorado logo and name. The footer contains navigation links: ABOUT, ACADEMICS, ADMISSIONS, STUDENT LIFE, ARTS, ATHLETICS, ALUMNI & GIVING, FOR YOU, and APPLY, along with a search icon.

Quality Control Check:

- It looks like the Description field in the 'Title and Metadata' section is missing. This field is important to help search engines find and understand what pages are about. Adding a description not only helps this page but pages that are linked to and from it. Please add a short sentence that describes the page and its purpose.
- Two links:
 - Go to the parameters of the page
 - [Learn more about Meta tags](#)

Code

```
12 <!-- QC checks! -->
13 <xsl:template name="quality-check">
14   <xsl:param name="document" />
15   <xsl:if test="$ou:action != 'pub'">
16     <xsl:variable name="meta" select="$document/ouc:properties[@label='metadata']/." />
17
18     <xsl:if test="$meta/meta[@name='Description'][@content='']">
19       <div class="alert-box alert qc-check">
20         <div class="row">
21           <p><strong>Quality Control Check:</strong> It looks like the Description field in the
22         </div>
23       </div>
24     </xsl:if>
25   </xsl:if>
26 </xsl:template>
27
28 </xsl:stylesheet>
29
```

It worked!

- Down to about 20% missing right now.
- Web Communications didn't have to do anything!
- Just the reminder helped web authors.



This also helped to remind us!

- We (administrators) jump in and out of pages quickly to fix 404 links, misspellings or emergency content updates.
- When we are in there fixing something we can also improve the SEO slightly by making up a description while we are in there.
- Nobody complained about the message.

Expanded to short meta descriptions

```
18 | <xsl:choose>
19 |   <xsl:when test="$meta/meta[@name='Description'][@content='']">
20 |     <div class="alert-box alert qc-check">
21 |       <div class="row">
22 |         <p><strong>Quality Control Check:</strong> It looks like the Description field i
23 |       </div>
24 |     </div>
25 |   </xsl:when>
26 |   <xsl:when test="string-length($meta/meta[@name='Description']/@content) lt 50">
27 |     <div class="alert-box warning qc-check">
28 |       <div class="row">
29 |         <p><strong>Quality Control Check:</strong> Thank you for adding a Descriptic
30 |       </div>
31 |     </div>
32 |   </xsl:when>
33 |   <xsl:otherwise></xsl:otherwise>
34 | </xsl:choose>
```

Questions
about
<title/> or
<meta/>?



Fixing HTML errors with XSL

This section is to protect you against the “dangerously advanced content author” who thinks they knows more about your design, your brand and HTML than you.

OUTC18

<template match="" />

- Using <template match="" /> you can match almost anything inside the XML nodes.
 1. <xsl:template match="table[@class='ou-content-slider']">...
 2. <xsl:template match="iframe[contains(@src, 'youtube')]">...
 3. <xsl:template match="p[ends-with(., '!!')]">...
 4. <xsl:template match="a[@id][not(@href)]">...

Remove Underline

- The underline toolbar button can be removed under 'Toolbars' but that doesn't prevent someone from using [Ctrl + U].
- The editor puts in a `` tag to create underline.
- It should be matched and removed:

```
9   <!-- remove underline -->
10  <xsl:template match="span[@style='text-decoration: underline;']" >
11  |   <xsl:copy-of select="node()" />
12  </xsl:template>
```

Removing `<p> </p>`

- Content Authors were using empty paragraphs to create space between things.
- We provided a class that added space for web authors and told them about it during training.

```
<xsl:template match="p[node() = '&nbsp;'] [not(*)]"></xsl:template>
```

What about removing Lorem Ipsum?

- No one would publish a page with Lorem Ipsum on it, right?



Demo text from sta...



Use this region for the mai...

High

17
Pages

```
<!-- removes paragraphs that start with 'Lorem ipsum' -->  
<xsl:template match="p[starts-with(., 'Lorem ipsum')]"></xsl:template>
```

Links to index.aspx (index.html)

- Our default page is index.aspx.
 - Yours could be index.php, index.html, index.htm, default.htm
- Duplicates the page with the directory
 - Unco.edu/jesse/index.aspx == unco.edu/jesse
- Keep the dependency tag within OU Campus to the file, but on publish, remove index.aspx.

Anchors where the href ends with 'index.aspx'

```
9  <!-- remove index.aspx -->
10 <xsl:template match="a[ends-with(@href, 'index.aspx')]">
11   <xsl:element name="a">
12     <!-- loop through link attributes -->
13     <xsl:for-each select="@*">
14       <xsl:attribute name="{name()}">
15         <xsl:choose>
16           <xsl:when test="name() = 'href'">
17             <xsl:value-of select="replace(., 'index.aspx', '')" />
18           </xsl:when>
19           <xsl:otherwise>
20             <xsl:value-of select="." />
21           </xsl:otherwise>
22         </xsl:choose>
23       </xsl:attribute>
24     </xsl:for-each>
25
26     <!-- display link text -->
27     <xsl:copy-of select="node()" />
28   </xsl:element>
29 </xsl:template>
```

Links with the text 'Click Here'

- Search for links on your site that contain only "Click Here"
- RegEx: `[C | c]lick [H | h]ere`

Options

Type Literal Text Regular Expression

Find `[C | c]lick [H | h]ere`



Not too bad, actually

Files Searched 17744
Files With Matches 33
Total Matches 34



Solution 1: Replace Click Here with URL

```
11 <xsl:template match="a[lower-case(node()) = 'click here']">
12   <xsl:element name="a">
13     <!-- copy all link attributes -->
14     <xsl:for-each select="@*">
15       <xsl:attribute name="{name()}">
16         <xsl:value-of select="." />
17       </xsl:attribute>
18     </xsl:for-each>
19
20     <!-- get the correct link text -->
21     <xsl:value-of select="@href" />
22
23   </xsl:element>
24 </xsl:template>
```

Solution 2: Use the Link's <title> tag:

```
38 <xsl:choose>
39   <xsl:when test="contains(@href, 'www.unco.edu') and ends-with(@href, '.aspx')">
40     <xsl:variable name="linkPath"
41       |   select="concat($ou:root, $ou:site, replace(replace(@href, 'http://www.unco.edu', ''), '.aspx', '.pcf'))" />
42     <xsl:variable name="linkTitle"
43       |   select="document($linkPath)/document/ouc:properties[@label='metadata']/title" />
44     <xsl:choose>
45       <xsl:when test="$linkTitle">
46         |   <xsl:value-of select="$linkTitle"/>
47       </xsl:when>
48       <xsl:otherwise>
49         |   <xsl:value-of select="@href" />
50       </xsl:otherwise>
51     </xsl:choose>
52   </xsl:when>
53   <xsl:otherwise>
54     |   <xsl:value-of select="@href" />
55   </xsl:otherwise>
56 </xsl:choose>
```

Solution 3: Remove the link and inform the user of the error.

```
62 <xsl:template match="a[contains(lower-case(node()), 'click here')] ">
63     <xsl:if test="$ou:action != 'pub'">
64         <span data-tooltip="data-tooltip" aria-haspopup="true" class="has-tip code-error"
65             title="The text 'click here' is not accessible for users with disabilities. Please rewrite
66             <!-- link text -->
67             <xsl:value-of select="node()" />
68         </span>
69     </xsl:if>
70 </xsl:template>
```



Content Region

Click Here Example.

Use this region for the main page content. Break paragraphs of text into reading-friendly co paragraphs. [Click here to learn more about people not clicking anymore !](#)

The text 'click here' is not accessible for users with disabilities. Please rewrite the link text without using the words 'click here'.

UNC 970-351-1890
501 20 ST.
GREELEY, CO 80639

MEDIA ABOUT UNC
UNC OVERVIEW
AWARDS & ACCOLAI

Use the same approach for ALT tags?

- OU Campus requires all images have an ALT tag.
- This doesn't prevent users from typing in 'alt', 'alt tag', 'ALT', etc.

Options

Type Literal Text Regular Expression

Find

Ignore Case

Replace With



Alt text with the word 'alt'



Action ▾

99.69%
COMPLIANCE



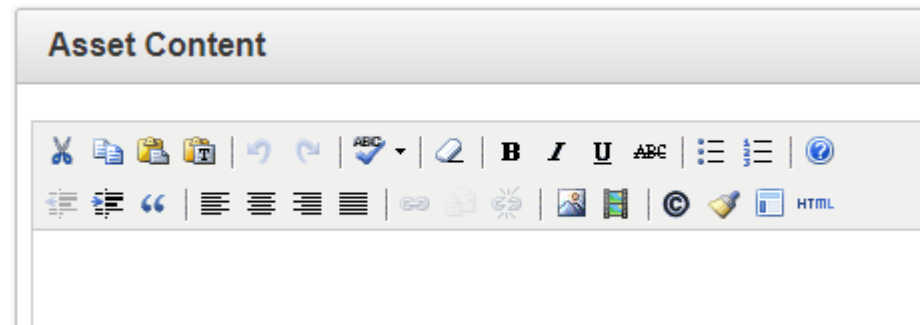
23
HITS

Finding ALT tags

- Find RegEx: `<img.*alt=((\".*\s)|\")[a|A]lt[\s|\"]`
 - Matches: `alt="alt *"` or `alt="*alt"` or `alt="alt"`
- Template Match suggestions:
 - `<xsl:template match="img[@alt='alt']">`
 - `<xsl:template match="img[contains(lower-case(@alt),'alt')]">`
 - `<xsl:template match="img[starts-with(lower-case(@alt),'alt ')]">`
- Combine?
 - `<xsl:template match="img[@alt='alt'] | img[starts-with(lower-case(@alt),'alt ')]">`

Remove <script>, <style>, or <link>

- The Web Content Asset allows a Content Author access to the HTML source of that asset.



- This can create problems when users think they can add <script/> to an asset and add it to a page

Remove <script>, <style>, or <link>

- Match any <script> tag inside of a user's editable region and replace with a comment.

```
9 <!-- Do not allow scripts in editable regions. This is mostly for the Web Content Asset: -->
10 <xsl:template match="ouc:div/script|ouc:div/*/script">
11   <xsl:comment>
12     Scripts in the content region have been disabled.
13     Contact Web Communications for help with executing JavaScript.
14   </xsl:comment>
15 </xsl:template>
```

Admins using: <script> & <style/>

```
<document xmlns:ouc="http://omniupdate.com/XSL/Variables">

  <headcode>
    <style type="text/css">
      .no-custom-style {margin-top:25px;}
    </style>
  </headcode> <!-- before closing head tag -->
  <bodycode></bodycode> <!-- after opening body tag -->
  <footcode>
    <script src="https://empoweredby.communityfunded.com/collaborate.js"></script>

    <script>
      $(document).ready(function(){
        if ( window.location.href.indexOf("cfpage=") !== -1){
          $(".images").hide();
          $(".intro-text").hide();
          $(".page-header").addClass("site-header-noimage UNC-blue minor-section").remove
        }
      });
    </script>
  </footcode> <!-- before closing body tag -->
```


Questions about
using template
matching & XSL
to remove
unwanted code
and stuff?



Structured Data, Open Graph

Hidden tags for robots that happens behind the scenes to help search engine optimization.

OUTC18

Submit Social Meta Information

- OmniUpdate provides example code on their GitHub: <https://github.com/omniupdate/example-code-social-meta>
- Example shown is what we use, and OmniUpdate's GitHub account has MUCH more!

Individual Page – Open Graph

```
10 <xsl:template name="social-meta">
11   <!--.opengraph start -->
12
13   <meta property="og:title" content="{ouc:properties/title/text()}" />
14
15   <meta property="og:url" content="{concat($domain, $ou:path)}" />
16
17   <meta property="og:description" content="{ouc:properties/meta[@name='Description']/@content}" />
18
19   <meta property="og:site_name" content="{document($path-props)/document/ouc:properties[@label='config']
20   /parameter[@name='section-title']}" />
21
22   <meta property="og:type" content="website" />
23
24   <!-- Add image, from our banner image slider -->
25   <xsl:if test="ouc:div[@label='banner-content']/table[@class='ou-banner-image']/tbody/tr[1]/td/img/@src !=
26   '' and ou:pcfparam('banner')='on'">
27     <meta property="og:image" content="{ouc:div[@label='banner-content']/table[@class='ou-banner-image']
28     /tbody/tr[1]/td/img/@src}" />
29   </xsl:if>
30 </xsl:template>
```

Result of OpenGraph Code

```
1 <!DOCTYPE HTML>
2 <html class="no-js" lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Graduate Programs</title>
7     <meta property="og:title" content="Graduate Programs">
8     <meta property="og:url" content="http://www.unco.edu/nhs/nursing/programs/graduate/index.aspx">
9     <meta property="og:description" content="graduate programs">
10    <meta property="og:site_name" content="Nursing">
11    <meta property="og:type" content="website">
```

Structured Data Resources

- Intro to Structured Data
<https://developers.google.com/search/docs/guides/intro-structured-data>
- Structured Data Testing Tools
 - <https://search.google.com/structured-data/testing-tool>

Bread Crumbs

- If your site has breadcrumbs, add a touch of extra code to submit these to Google.
- Documentation:
<https://developers.google.com/search/docs/data-types/breadcrumb>

Graduate Programs - University of Northern Colorado

www.unco.edu › [College of Natural and Health Sciences](#) › [Nursing](#) › [Programs](#) ▼

Sep 29, 2017 - **Graduate Programs**. Master Degree Programs. Master of Science: Nursing (MSN) Adult-Gerontology Acute Care Nurse Practitioner (AGACNP) Emphasis · Master of Science: Nursing (MSN) Family Nurse Practitioner (FNP) Emphasis ...

_shared/breadcrumbs.xsl

- If your instance is like mine, you should have a file in your _shared folder called breadcrumb.xsl
- This file pulls in information from the _props.pcf file in each folder/section and builds out the Breadcrumbs

Nursing

GRADUATE PROGRAMS

[Home](#) [About Us](#) ▾ [Programs](#) ▾ [Resources](#) ▾ [FAQs](#)

[UNC](#) > [College of Natural and Health Sciences](#) > [Nursing](#) > [Programs](#) > [Graduate Programs](#)

HTML example of Breadcrumbs:

```
▼ <ul class="breadcrumbs" vocab="http://schema.org/" typeof="BreadcrumbList">
  ▼ <li property="itemListElement" typeof="ListItem">
    ::before
    ▼ <a property="item" typeof="WebPage" href="/">
      <span property="name">UNC</span>
      <meta property="position" content="1">
    </a>
  </li>
  ▼ <li property="itemListElement" typeof="ListItem">
    ::before
    ▼ <a property="item" typeof="WebPage" href="/nhs/">
      <span property="name">College of Natural and Health Sciences</span> ==
      <meta property="position" content="2">
    </a>
  </li>
  ▼ <li property="itemListElement" typeof="ListItem">
```

Breadcrumbs XSL additions

- Called within a template called bc-folders
- Position is left blank and filled later with JS.

```
103 <!-- Previous List item:
104 <li><a href="{concat($link-start,$path)}"><xsl:value-of select="$title"/></a></li>
105 -->
106 <!-- Structured Data List Item -->
107 <li property="itemListElement" typeof="ListItem">
108     <a property="item" typeof="WebPage" href="{concat($link-start,$path)}">
109         <span property="name"><xsl:value-of select="$title"/></span>
110         <meta property="position" content="" />
111     </a>
112 </li>
```

Breadcrumbs jQuery

- A bit of jQuery to populate the position attribute

```
4     $.fn.StructuredDataBreadcrumbs = function(){
5         var i = 1;
6         $("ul.breadcrumbs li").each(function(){
7             $(this).find("meta").attr("content", i);
8             i++;
9         });
10    };
```

Results (overtime)

- Structured Data in Google Webmaster Tools
- I didn't want to republish the whole site.

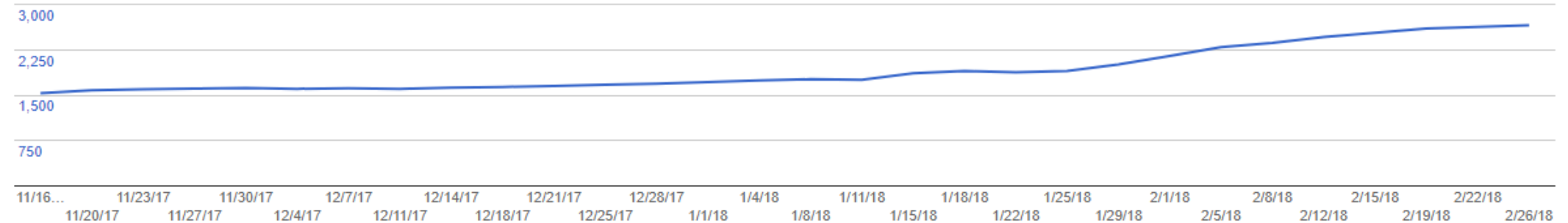
Structured Data

Status: 2/26/18

■ 2,656 Items ?
on 2,653 pages

0 Items with Errors ?
on 0 pages

Items



Questions
about
Structured
Data, previous
questions or
anything else?





Dream.



Discover.



Develop.

OUTC18

Thank you.

Jesse.Clark@unco.edu

[@JesseClark](#)

JessClark.com

Unco.edu

outc18.com/surveys