



*A Data Usability Company*  
**ANTENNA HOUSE**

# Loose-leaf Formatting with CSS

XML Prague 2020

**Tony Graham**

XML Division

Antenna House, Inc.

tgraham@antenna.co.jp

tony@antennahouse.com

@tgraham\_antenna

# Loose-leaf Formatting with CSS

- What is loose-leaf formatting?
- Why use loose-leaf?
- Municode project
  - Editor customization
  - Schematron
  - XSLT
  - Making pages

# What is loose-leaf formatting?

- Printed document maintained as pages (leaves) in a binder
- On update:
  - Remove old version of changed pages
  - Insert new version of changed pages
- What happens when an update changes pagination?

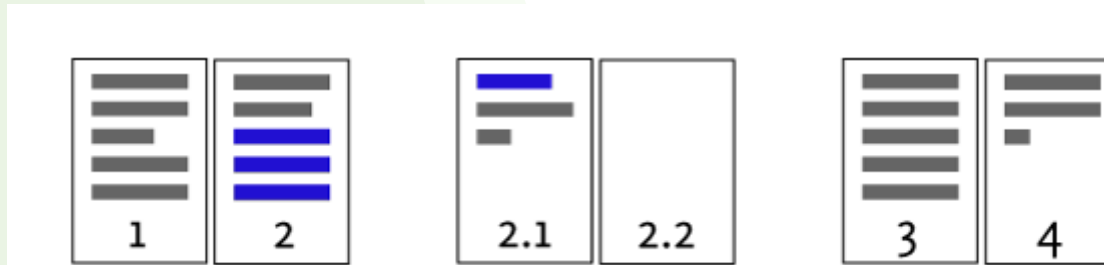
# Initial version

- Initial document has no changes



# Update adds pages

- Update may add text



- Distribute only changed pages ('take')



# Update adds and changes pages

- Update may add and change text



- Distribute only changed pages



# General Logic for Paginating Takes

- Point pages should be avoided as much as possible.
- If new point pages are needed, it is preferred to add them at the END of take.
- If next page in the book is a point page ending with .1, then any new point pages should be added at BEGINNING of take rather than at the end.
- As a last resort, add point pages anywhere in the MIDDLE of a take.



- Provides codification (and other) services to municipalities
- Replacing existing loose-leaf system
  - More automation
  - Author as HTML5 for reuse on web
  - CSS formatting using AH Formatter
  - Initial development by Eliot Kimber





# “Loose-leaf publishing using Antenna House and CSS”

- Eliot's talk at Balisage 2019
- <https://www.balisage.net/Proceedings/vol23/html/Kimber01/BalisageVol23-Kimber01.html>
- <https://www.slideshare.net/drmacro/loose-leaf-publishing-using-antenna-house-formatter-and-css-for-pagination>

# Why loose-leaf?

- Municipal codes updated frequently
  - Potentially every council meeting
- 1,000+ pages
- Staff used to working with paper copies
- Impractical to reprint on every update
- Print and distribute only changed pages

# Markup

- Controlled subset of HTML5
- data-\* attributes for aspects of formatting
- `<?pdf?>` processing instructions
  - Mark start and end of changed region ( 'take' )

## data-\* attributes

- `<p data-type="subt i l e" >`  
Type information used by XSLT and CSS
- `data-break-before="ri ght"`  
`data-break-i nsi de="avoi d-page"`  
`data-break-after="col umn"`  
`data-pgl ayout=' narrow'`  
Formatting overrides

# <?pdf?>

- Mark start and end of a 'take'
- Added manually
- Occur almost anywhere
  - Between elements
  - Between <tr> but not within a table cell
  - Inside text
  - Inside a word

## <?pdf?> inside a word

```
<?pdf take-begin job="S001-U01"
firstpage="9"?>
```

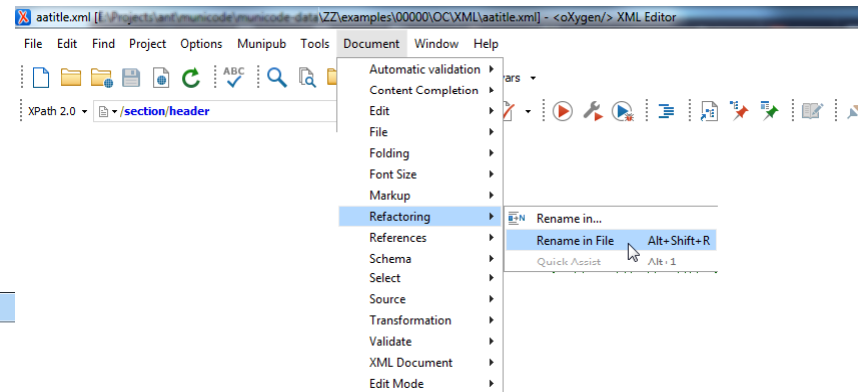
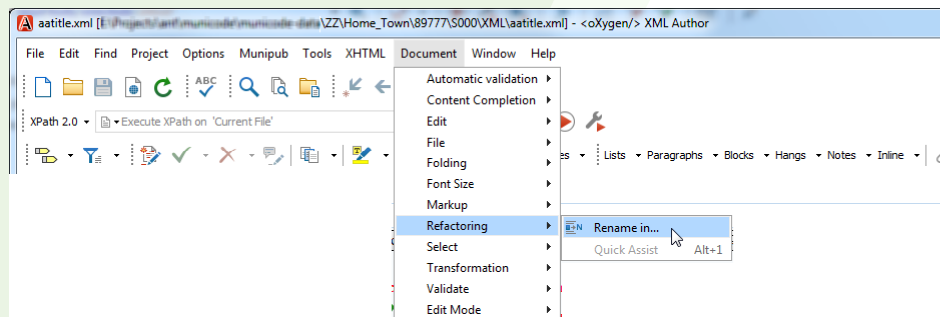
```
<p>[TEXT]</p>
```

```
<p>Addit<?pdf take-end
job="S001-U01" firstpage-ref="9"
textafter="-"?>ional text inside
a paragraph.</p>
```

# Editor

- Editors
  - Oxygen XML Author
  - Custom toolbar
  - Limited menus
  - Custom 'Munipub' menu

- Developers
  - Oxygen XML Editor
  - Full Oxygen menus



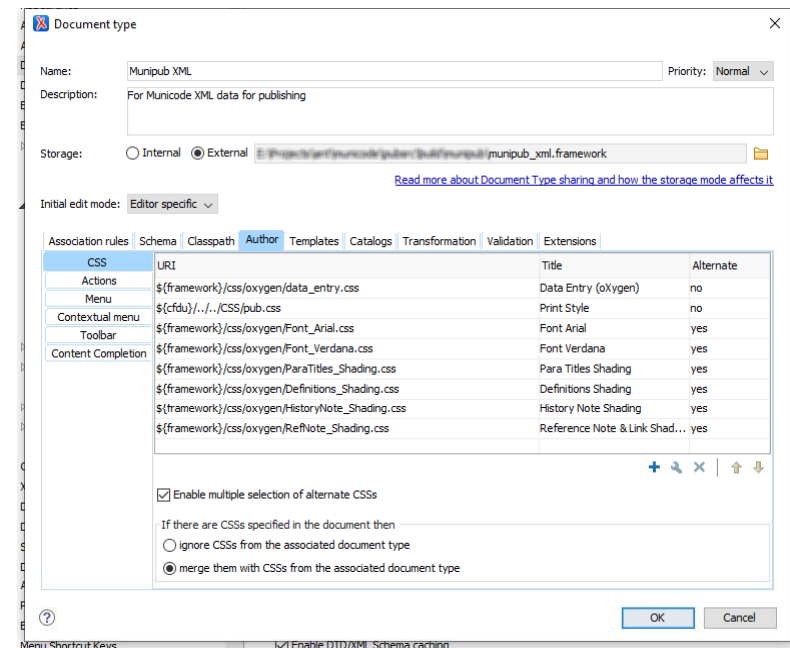
# CSS: yours, mine, and ours

- Oxygen-specific CSS used only in Oxygen
- AH Formatter extensions used in print stylesheets
- Common CSS used by both



# Oxygen-specific CSS in framework

- Oxygen-specific CSS used only by Oxygen framework



# Oxygen with non-Oxygen CSS

- Syntax-aware editor should warn of errors
- Can't make Oxygen recognise extensions

```
71 @sidenote {
72     /*counter-increment: sidenote;*/
73     float: outside;
74     clear: both;
75     width: 20%;
76 }
77
78
79
80
```

**Validation:**

! "@sidenote" is a CSS hack : @sidenote { /\*counter-increment: sidenote; \*/ float: outside; clear: both; width: 20%; }

Press F2 for focus

In order to express sidenotes by CSS, (-ah-)float: sidenote and @sidenote are used. The same way as footnotes applies to sidenotes. [\[CSS3-GCPM\] Sidenotes \(@sidenote has been removed from GCPM on and after June 8, 2010\)](#)

```
span.sidenote {
  -ah-float: sidenote;
}
@page {
  @sidenote {
    -ah-float: outside;
    clear: both;
    width: 20%;
  }
}
<p>
Lorem dignissim<span class="sidenote">Quisque suscipit ante vel
eros.</span>, orci ac porta blandit, ...
</p>
```

# Schematron: data-\* attributes

- data-\* attribute values affect formatting
- data-\* attributes required in some contexts

```

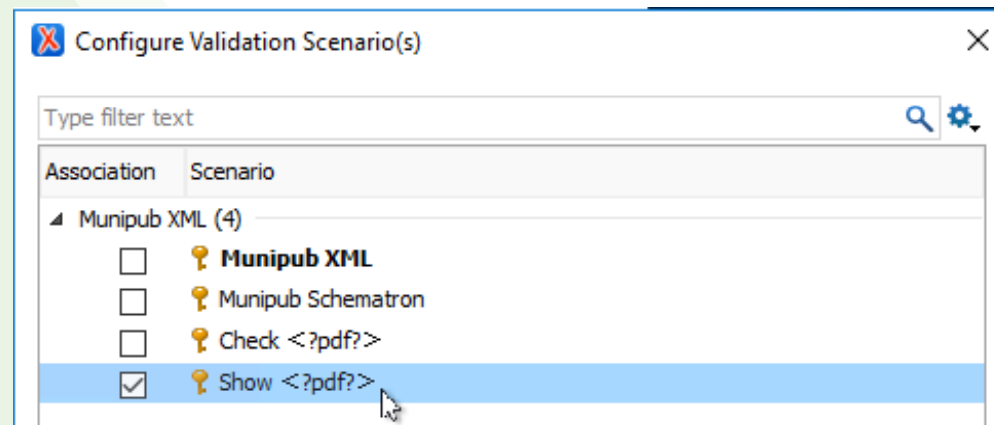
<rule context="/xh: section/xh: header">
  <report
    test="empty(xh: p[@data-type = 'startpage']) or
         normalize-space(xh: p[@data-type[. = 'startpage']]
                           = ''"
    sqf: fix="header-add-startpage"
    role="warning">Section does not have a 'startpage'. A value
of '1' will be assumed.</report>
  <sqf: fix id="header-add-startpage">
    <sqf: description>
      <sqf: title>Add a 'startpage' paragraph.</sqf: title>
    </sqf: description>
    <sqf: add position="last-child">&#xA;</sqf: add>
    <sqf: add position="last-child">
      <xh: p data-type="startpage">1</xh: p>
      <xsl: text>&#xA;</xsl: text>
    </sqf: add>
    <sqf: add position="last-child">&#xA;</sqf: add>
  </sqf: fix>
</rule>

```

# Schematron: <?pdf?>

- Pseudo-attribute values
  - saxon: get-pseudo-attribute()
- Interrelated pseudo-attribute values
- Correspondence between start and end <?pdf?>
  - Final end <?pdf?> may be omitted
  - Start <?pdf?> must have higher page number than previous

# 'Budget' PI locator



section processing-instruction

**State law reference(s)**—Cities and municipalities generally, K.S.A. 12-101 et seq.; ordinances generally, K.S.A. 12-3001 et seq.; compilation, revision and codification of city ordinances, K.S.A. 12-3014 et seq.; incorporation of subjects by reference, K.S.A. 12-3009 et seq.

`mcc minitoc;section;nopg`

`pdf take-begin job="S001" firstpage="24;3" nextpage="25"`

Elements

**Validation:**

`<?pdf take-begin job="S001" firstpage="24;3" nextpage="25"?>`

Press F2 for focus

# 'Budget' PI locator

- Every PI generates an 'info' report

```
<phase id="pdf-pi-show">  
  <active pattern="pdf-pi-show-pattern"/>  
</phase>
```

```
<pattern id="pdf-pi-show-pattern">  
  <rule context="processing-instruction('pdf')">  
    <report test="matches(., '(^|\W)take-(begin|end)\W')"  
      role="info">  
&lt; ?pdf <value-of select="."/ >?</report>  
    </rule>  
</pattern>
```

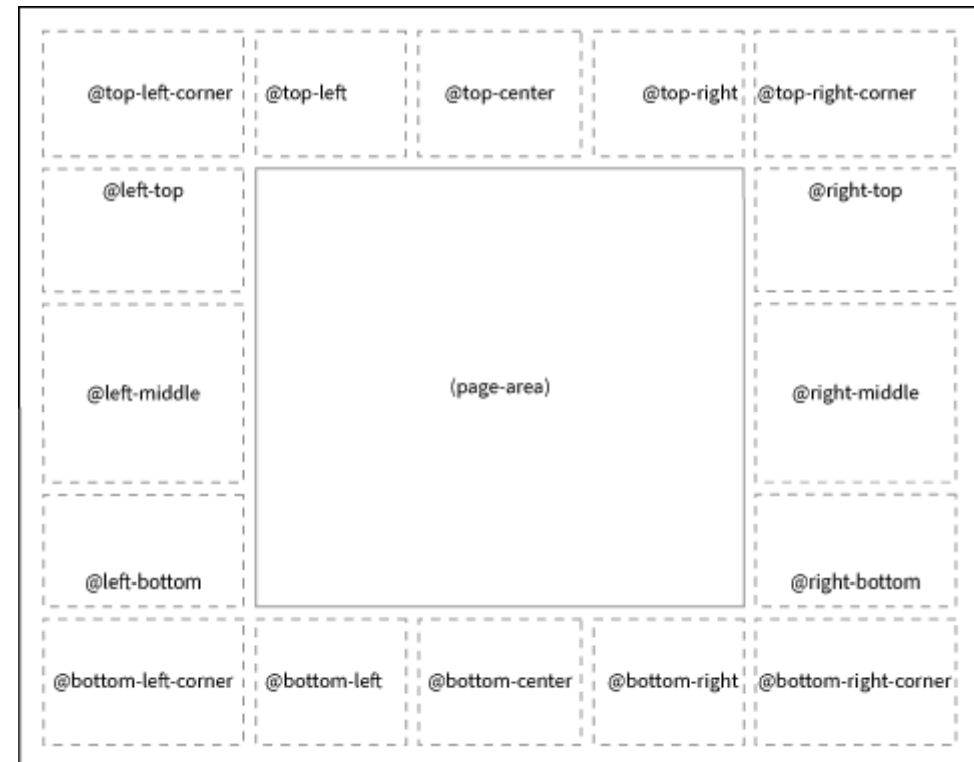
# Formatting

1. Transform HTML for easier formatting
2. Format to generate Area Tree XML
3. Adjust page numbers in Area Tree XML
4. Potentially reduce to just 'takes'
5. Format Area Tree XML to PDF
6. Update master record of page numbers



# Content in page-margin boxes

- Headers and footers
- Information hidden in 'unused' boxes
  - Page number format
  - Debug information



# Municode page numbers

- Municode page number may have per-chapter 'prefolio' and 'postfolio':
  - E.g. "CH-1 2 –"
  - Per-chapter numbering simplifies loose-leaf
- Different page formats in ToC, etc.
  - 1, i, I, a, A

# Problem with CSS page numbers

- CSS 'page' counter just a well-known counter
- Reference needs to know counter format  
target-counter()  
= target-counter( <url> , <custom-ident>  
[ , <counter-style> ]? )
- 'Prefolio', number format, and 'postfolio' in page-margin boxes for later processing

# Page layouts

- Portrait – 8.5x11
- Narrow – 8.5x11, narrow margins
- Landscape – 8.5x11, rotated
- Tabloid – 8.5x14
- Some data-type select a page layout
- data-pgl ayout overrides default

# data-type=" . . . "

```
/* These should only be part pages */
section[data-type = 'part'],
section[data-type = 'titlearr']
{
    counter-reset: page 1;
    break-before: right;
    page: part portrait;
}

section[data-type = 'part'][data-part-type = 'page-number']
{
    page: part-first-page-number portrait;
}
```

# data-pgl ayout=" . . . "

```
div[data-pgl ayout = "tabl oi d"],  
secti on[data-pgl ayout = "tabl oi d"] {  
    page: tabl oi d !important;  
}
```

```
/* The first tabl oi d page must start on a right-hand page. */  
*[data-pgl ayout = "portrai t"] + *[data-pgl ayout = "tabl oi d"],  
*[data-pgl ayout = "l andscape"] + *[data-pgl ayout = "tabl oi d"],  
*[data-pgl ayout = "narrow"] + *[data-pgl ayout = "tabl oi d"] {  
    break-before: ri ght;  
}
```

# @page rules

@page  
@page :left  
@page :right  
@page :blank  
@page portrait:first:right  
@page portrait:left  
@page portrait:right  
@page portrait:blank  
@page update-ins:first:right  
@page update-ins:left  
@page update-ins:right  
@page update-ins:blank  
@page toc-main:blank, lof-main:blank, lot-main:blank  
@page toc-main, lof-main, lot-main  
@page toc-main:first, lof-main:first, lot-main:first  
@page toc-main:left, lof-main:left, lot-main:left  
@page toc-main:right, lof-main:right, lot-main:right  
@page landscape  
@page landscape:left  
@page landscape:right  
@page landscape:blank

@page tabloid  
@page tabloid:left  
@page tabloid:right  
@page tabloid:blank  
@page narrow  
@page narrow:left  
@page narrow:right  
@page narrow:first:right  
@page narrow:blank  
@page frontmatter  
@page frontmatter:first  
@page frontmatter:blank  
@page officials  
@page officials:blank  
@page titlepage:first  
@page titlepage  
@page part:first  
@page part-first-page-number:first  
@page index:first:right  
@page index  
@page index:left  
@page index:right  
@page index:blank

# @page rule

```
@page {  
  size:      8.5in 11in;  
  margin-left: 7.5pc;  
  margin-right: 7.5pc;  
  counter-reset: footnote;  
  
  @bottom-left-corner {  
    content: ' ^npm^' counter(page);  
    visibility: hidden;  
    font-size: 8pt;  
    color: white;  
    /*color: red; */  
    font-family: monospace, 'Arial Unicode' ;  
  }  
}
```



# Conclusion

- Loose-leaf formatting is challenging
- HTML+CSS make authoring easier
- Some CSS aspects make loose-leaf harder
  - Reconstructing page number
- XSLT and Area Tree XML make it tractable
- Had to exploit 'vacant' page-margin boxes



*A Data Usability Company*

**ANTENNA HOUSE**

# Thank you

**Tony Graham**

XML Division

Antenna House, Inc.

tgraham@antenna.co.jp

tony@antennahouse.com

@tgraham\_antenna