ORACLE

Preprocess? Preprocess2?

Tales of an HTML5 plugin upgrade

Robert D Anderson

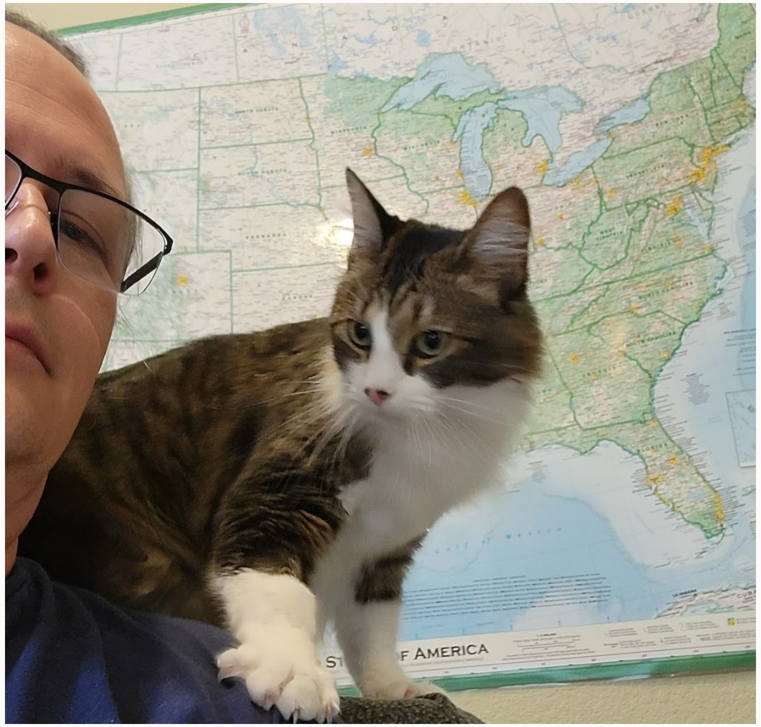
DITA-OT Day 2024
Helsinki

Agenda

- A bit about our plugin
- Why update?
- How it went
- Questions / Resources

Introduction – why are you hearing from me?

- Robert D Anderson
- Been working on DITA and DITA-OT related tools for a long time
- Editor of OASIS DITA Specification, Versions 1.1, 1.2, 1.3, and now 2.0



Target audience

- Anyone with a custom HTML5 plugin, that calls out to "dita2html5"

- Anyone who customizes "dita2html5" and calls "preprocess"

What does our plugin do (the processing part)?

- Mostly style related
- But we have customized preprocess
  - Message customization
  - Attribute cascading
  - Metrics report
  - Topicpull:
    - Customize for "dropdown" element
    - Ignore terms that become links
    - Hotfixes
  - Tweak maps ahead of chunking
  - Several "preprocess.post" steps

How would I describe preprocess?

A bit confining

Sort of spaghetti-ish

Why did I use preprocess to begin with?

We started with DITA-OT 3.4

Preprocess2 was there, but not default

Built "oraclehtml5" but … stuck with defaults

Perhaps a good reason to switch the default

Upgrade is easier if…

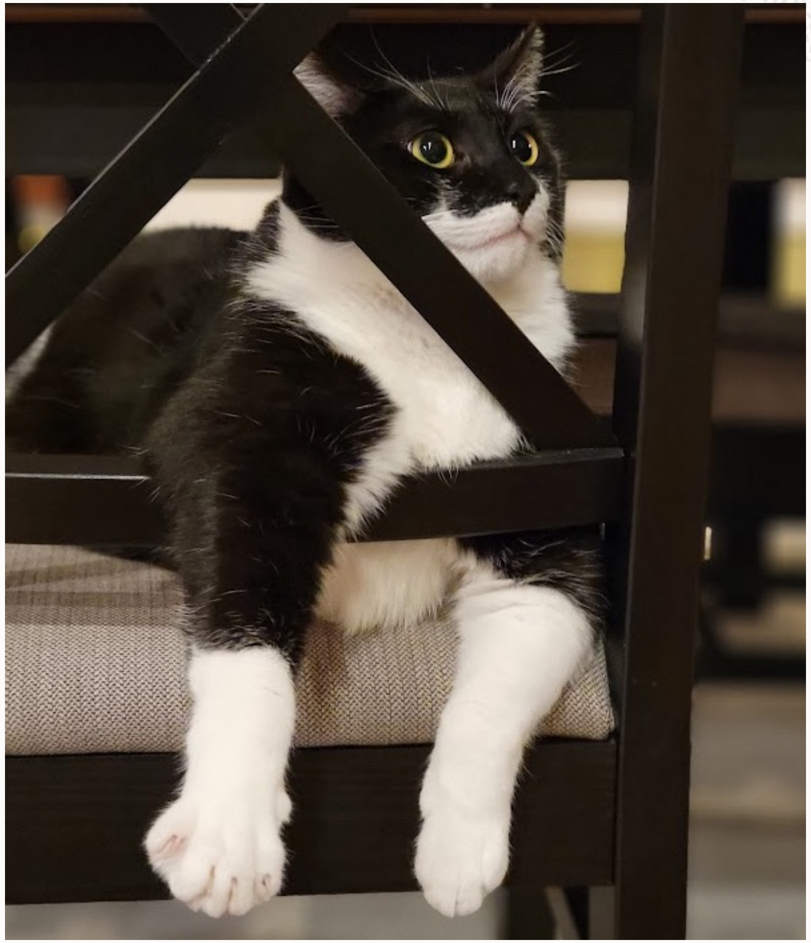
- You have a recent DITA-OT

- You have test cases for your overrides

This is effectively a migration, you want to make sure before/after is equivalent

First steps

- Go to latest toolkit
  - If you have a lot of customizations, maybe use 4.1.2
- In "dita2html5" Ant target, add a 2
- See if tests work



Result

…they didn't

If they work for you, proceed to the end, "Integration testing"

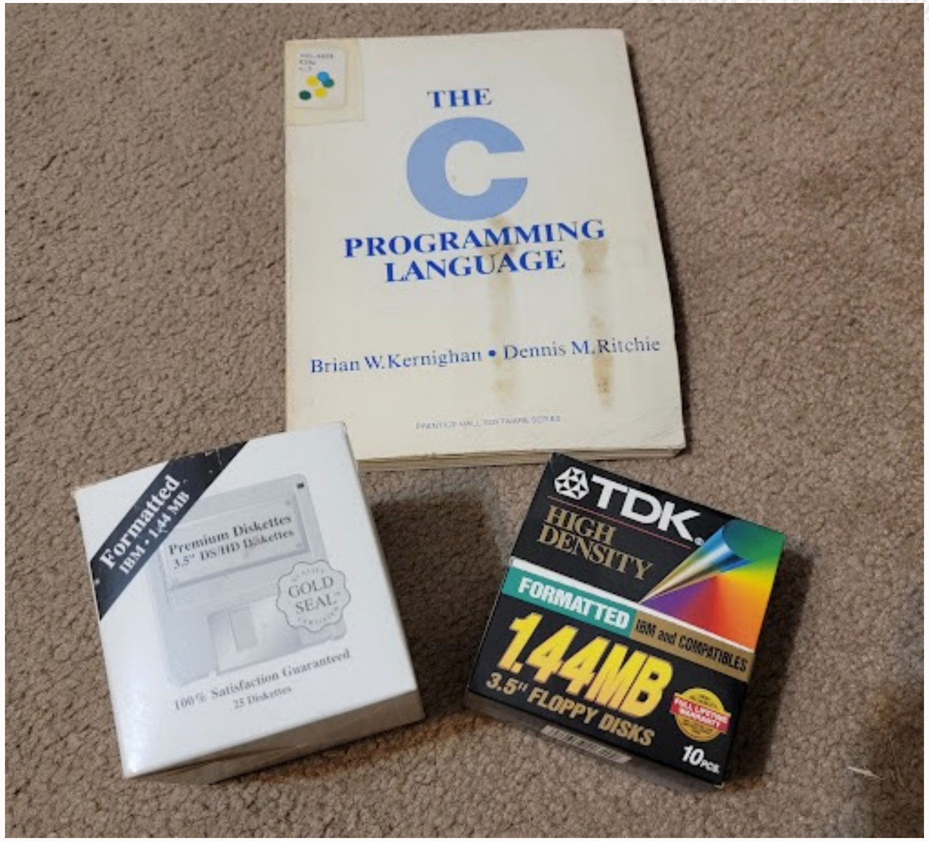
So, we have updates. Where to start?

This is a great opportunity to remove technical debt!

```
<!-- Provides a temporary fix for
  https://github.com/dita-ot/dita-ot/issues/3582

  Can be removed if upgrading to a release
  that includes that fix (likely 3.6 or greater). -->
```

...Remember, tests are your friend


THE C PROGRAMMING LANGUAGE
Brian W. Kernighan • Dennis M. Ritchie
Formatted IBM • 1.44 MB
Premium Diskettes
GOLD SEAL
100% Satisfaction Guaranteed
TDK
HIGH DENSITY
FORMATTED IBM and COMPATIBLES
1.44MB
3.5" FLOPPY DISKS
10PCS.

Tests still fail, so…

Time to build my own "map first processing"

Replace "dita2html5" with "ocidita2html5":

```
<target name="dita2html5"
    depends="html5.init,
    build-init,
    preprocess2,
    html5.topic,
    html5.map,
    html5.css"/>
```

Integrate check build, see what works…

Tests still fail, so…

Time to build my own "map first processing"

Replace "dita2html5" with "ocidita2html5":

```
<target name="ocidita2html5"
    depends="html5.init,
    build-init,
    ocipreprocess2,
    html5.topic,
    html5.map,
    html5.css"/>
```

Integrate check build, see what works…

XSLT extensions

…still work

```
<feature extension="dita.xsl.topicpull"
         file="xsl/topicpull-dropdown.xsl"/>
<feature extension="dita.xsl.conref"
         file="xsl/oci_conref.xsl"/>
```

Same XSLT is called; XSL imports are still there

…but I'll be able to get rid of that conref one with 4.2

Preprocess-pre, and preprocess-post

```
<feature extension="depend.preprocess.pre"
         value="oracle.preprocess.pre"/>

<feature extension="depend.preprocess.post"
         value="oracle.preprocess.post"/>
```

Also still work. Always good to use these if possible.

On to the tricky ones

```
<feature extension="depend.preprocess.chunk.pre"
         value="generate-metrics-before-chunk"/>
<feature extension="depend.preprocess.chunk.pre"
         value="oci-chunk-pre"/>
<feature extension="depend.preprocess.move-meta-entries.pre"
         value="oci-movemeta-pre"/>
```

(Should I override topic-chunk, etc?)

Add in between "map-first" and "then-topics"!

Chunk processing happens in "preprocess2.topics"

```
<target name="preprocess2"
  ...,

  preprocess2.maps,
  preprocess2.topics,
```

Add in between "map-first" and "then-topics"!

Chunk processing happens in "preprocess2.topics"

```
<target name="ocipreprocess2"
  ...,

  preprocess2.maps,
  generate-metrics-before-chunk,
  oci-chunk-pre,
  preprocess2.topics,
```

- Keeps updates isolated
- Minimize changes within "maps" or "topics"

And finally, insert in my custom preprocess2.maps

I *thought* I would need something like this:

```
<target name="ocipreprocess2.maps"
  depends="map-reader,
  map-mapref,
  map-branch-filter,
  map-keyref,
  map-conrefpush,
  map-conref,
  oci-add-my-target-here,
  map-profile"/>
```

But what I actually needed was…

- I need to run a step before metadata cascades
- But … that cascade is also in preprocess2.topics
- So that also can go in between:

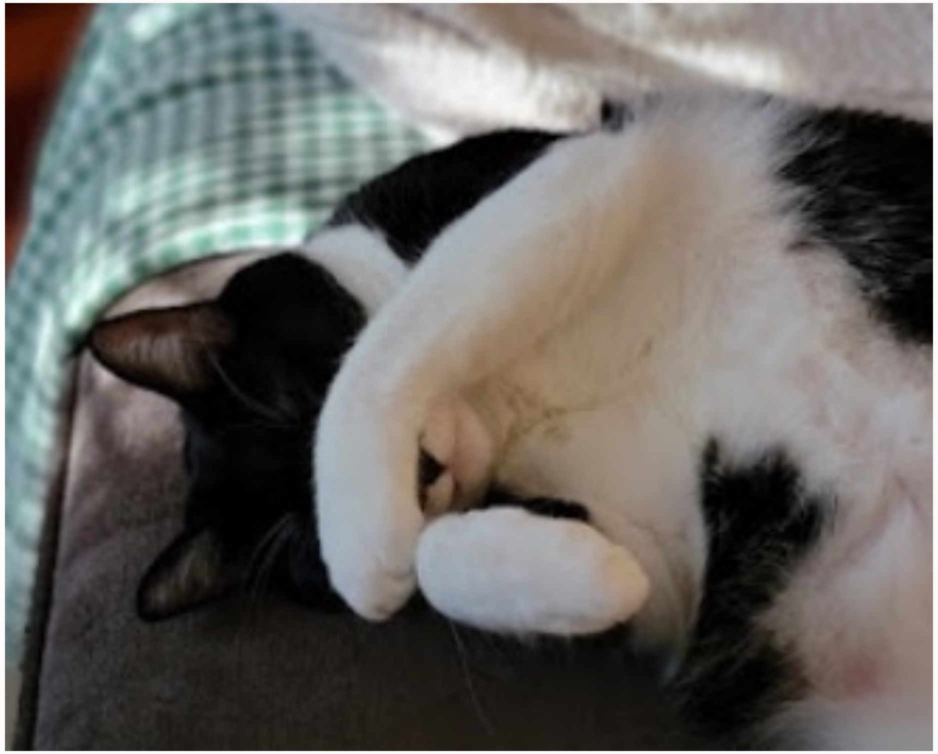
```
<target name="ocipreprocess2"
  ….,

  preprocess2.maps,
  generate-metrics-before-chunk,
  oci-chunk-pre,
  oci-movemeta-pre,
  preprocess2.topics,
```

What's left?

- Wait, is that all?
  - Chunk and metadata extensions: in between "preprocess2.maps" and "preprocess2.topics"
  - XSLT extensions: no impact
  - HTML5 / style customizations: no impact
- Do I even need to override the "map first" or "topic later' sections?

- ***… no***



Final result:

```
<target name="ocipreprocess2"
  dita:extension="depends org.dita.dost.platform.InsertDependsAction"
  description="Preprocess"
  dita:depends="{depend.preprocess.pre},
  preprocess2.init,
  ditaval-merge,

  preprocess2.maps,
  generate-metrics-before-chunk,
  oci-chunk-pre,
  oci-movemeta-pre,
  preprocess2.topics,

  map-clean-map,
  clean-preprocess,
  copy-files2,
  {depend.preprocess.post}">
```

Test test test

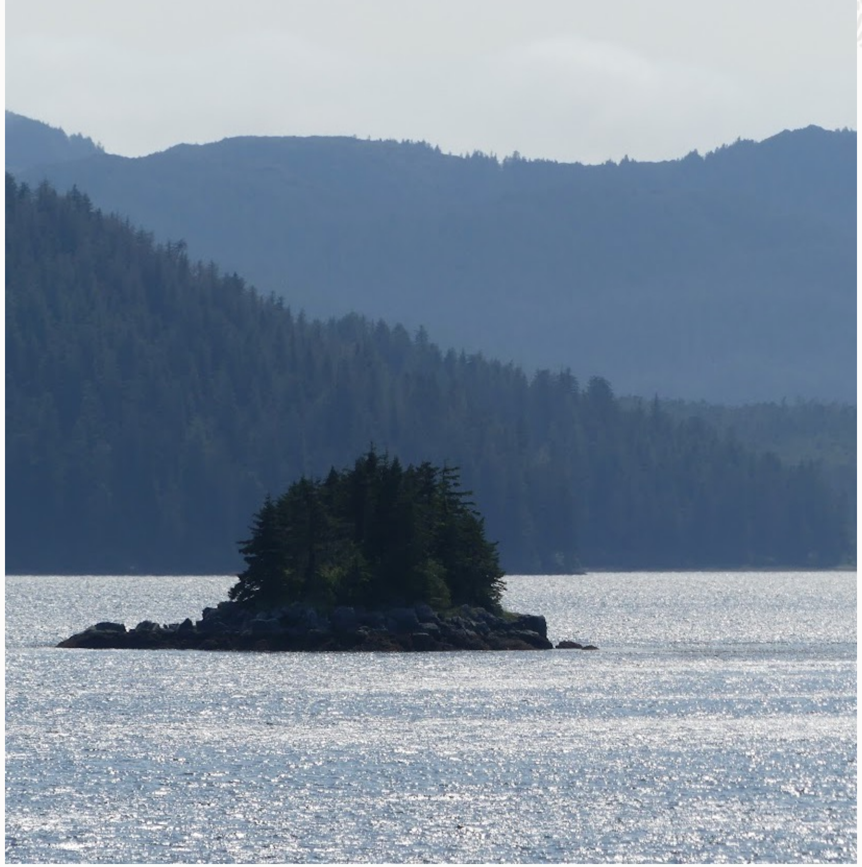
Full steam ahead! With integration testing

Integration testing: results

- Image file paths were condensed!
  `"../../images/service1/../service2/some.png"`
  ➔
  `"../../images/service2/some.png"`

- Unique IDs changed –
  `id="unique_1129213513"`
  ➔
  `id="unique_123292114"`

- Found one bug that's easy to workaround: https://github.com/dita-ot/dita-ot/issues/4371



What if you did this, and it didn't work?

- Follow steps like I outlined above with "ocidita2html5"...
- ...but keep calling "preprocess"
- Open a bug report for issues you encountered
- Once fix is available, you're ready to upgrade

Going forward

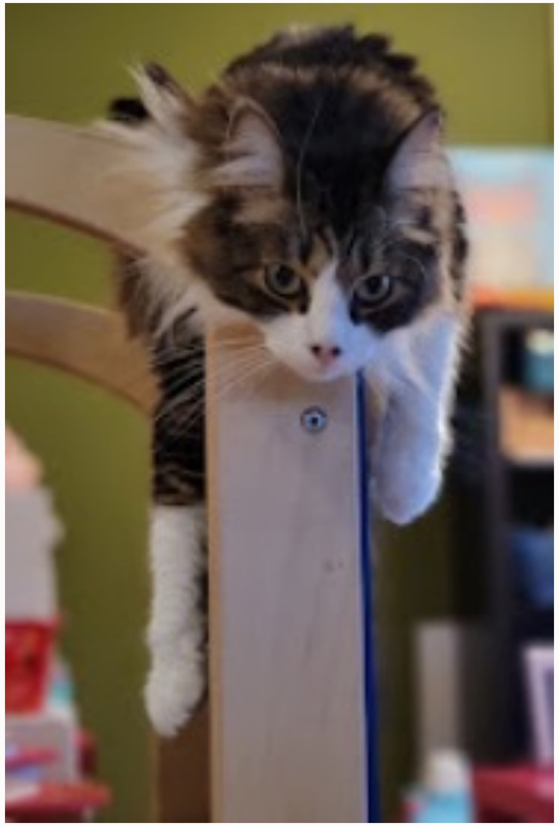
I do have a new dependency – I have to watch for updates to preprocess2

But only in that single target, which is unlikely to change often

And I don't have to care what else gets added or moves around

Summary

- Be on a recent DITA-OT
- Clear out tech debt if you can
- Be sure you have good unit tests for your customizations
- Add a 2 in the Ant build, re-integrate, and test
- Update where needed
- Don't do more than you need to
- Integration test with full real-world content
- Remember to watch for "preprocess2" changes going forward



Thank you!