Structuring for Content Reuse with Doc·To·Help

MadCap Webinar – February 1, 2017
Hello!

I am Mary Connor

I am here because I love helping fellow tech writers survive and thrive.

You get cats, too. You can find me at [www.cleverhamster.com](http://www.cleverhamster.com)
But first, let’s deal with this:
“You’re using WORD for documentation?
Get a REAL tool!”
~ Says everybody, everywhere
So, why Doc-To-Help?

Single-sourcing across targets and sources: HTML, XHTML, DOCX

One strategy for user and API docs

Team features: source files in TFS, SharePoint; familiar editors, no markup

Command-line builds, build scheduler

Able to handle our immense builds

Affordability
Caveat: Be sure to disable the “newbie” features!
...and set up for serious global editing:
Our Scope: Soup to Nuts!

1. Standardize it
2. Structure it
3. Reuse it
Standardize It
Merge and Purge
“IT’S ALL IN THE PREP!”

1. **Unify** content into one DOCX
2. **Purge**, scrub, clean, trim
3. **Standardize** heading structures

… *before* you divide for reuse
1- UNIFY
Case Studies:

- Author-it to Doc-To-Help
- DocBook XML to Doc-to-Help

...which means, yes, you really can!

Is this even possible?
Migrating from a CCMS*

- Unraveled existing reuse so each “object” publishes just once
- Built new book to export all to a single, linked DOC
- Optimized templates for D2H import with Word macros to post-process: resizing graphics, substituting styles, deleted section breaks, etc.
- Automated migration so I could iterate testing (speeded up project hugely)

*component content management system
Migrating from DocBook

◎ Used **xmlint** to process all XML projects (several PDFs per product) into master XML source (**merge.xml**)

Used **Pandoc** transform tool to build merged XML into master Word file

```
REM  Project to export Docbook to Word, February 2015
REM  Pandoc for doc conversion; see http://johnmacfarlane.net/pandoc/README.html

cd C:\svn\Swarm\Docbooks\CASTorAdvancedAdministrationGuide
xmlint --xinclude CCASTorAdministrationGuide.xml --output merged.xml
pandoc -s -f docbook -t docx -o CCASTorAdministrationGuide.docx merged.xml
pandoc -s -f docbook -t markdown -o CCASTorAdministrationGuide.txt merged.xml
```
2- Purge! :-) 

- Clear section/page/line breaks
- Trim wordiness
  - “For more information about blah, refer to” “See”
- Legacy handholding
  - (e.g., browsers aren’t new)
- Useless screenshots (no data)
- Metadiscourse (“As discussed”)
- Move out “touch-once” content
Survival: Do more with less.

1. **Do you really know what users need?**
   - Yes: **Can you skip maintaining it?**
     - Yes: Create touch-once docs
       - Tech Note/Bulletin
       - Blog, Support forum
       - KB Article
       - Searchable docs/KB
     - No: Determine their actual needs
       - Support case mining
       - Sales engineers
       - Tech trainers
       - User survey
   - No: **Can it be automated?**
     - Yes: Get tools and create it
       - Contextual help
       - Generated/API doc
       - Use code comments
       - Embed external files
     - No: **Can it be visual?**
       - Yes: Get tools and create it
         - Screencam
         - Annotated screenshot
         - Diagram
         - Matrix
       - No: Create JBGE official docs
         - "Just Barely Good Enough" *

* from Disciplined Agile Consortium, Scott Ambler
...and scrub for consistency

- Use Word’s **style** checker!
- Enable everything
- Set per your style guide (commas, spaces)
- Improve readability for i18n, translatability (passives, noun strings)
3- Standardize Headings

- Fix levels (super easy in Outline)
  - Equalize, flatten (promote)
  - Convert excess to subheads
- Fix wording, by level (iterate)
  - H1 Product Name
    - H2 Noun (area)
    - H3 Gerund (task)
- Fix repetition, eliminating as can
Make Headings Orderly, Unique, and Standalone!

Fix h1 level, then h2, ...
Topic-Based Writing: Modularized Chunks

- Similar goals to DITA *
- **Standalone** chunks (final division)
- **Unified** per Minimalism
  - Includes what ya need to succeed
  - Action-oriented on user goals
  - Troubleshooting, recovery
  - Findability without linking away

Topic = **RLO**
(from instructional design)

“A **Reusable Learning Object** is ... a digital educational **resource (chunk)** that can be reused, scaled, and shared from a **central online repository** in the support of instruction and learning.”

**Relevant traits:**
- Standardized
- Self-contained
- Reusable
- Combine them into courses (deliverables)

How do I reuse content for docs *and* training?

http://ecolearnit.ifas.ufl.edu/concept.asp
Encapsulation by RLO

Approach (think, "module-only")

1. Reuse is only by RLO
2. RLO = source file
3. Only divide if required for reuse across outputs
4. Organize away most links
5. If dividing is getting hard or messy, reorganize the content
RLO rules, AIT

1. All content must be managed in a container/map/book
2. All content reuse occurs via container
3. Build outputs are comprised of containers
Prepare to break by chunk (RLO), here by area + audience

# = audience
0 = all, prospect, mkt
1 = cheap license
2 = staff license
3 = admin access
4 = developer
1. Create a DOCX in D2H
2. Clone it $n$ times
3. Rename per scheme
4. Copy/paste content from Outline to matching file
1. Make a copy
2. Attach D2H template
3. Use **Master Document** to quickly **Create** subdocuments
...Then have D2H import the entire folder

Remember: Keep filenames team-friendly; they don’t appear in outputs

Powerful flexibility!
You can build immediately!

<table>
<thead>
<tr>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>a0-Help and Support.docx</td>
</tr>
<tr>
<td>a2-Keyboard Shortcuts.docx</td>
</tr>
<tr>
<td>a0-Billing Overview.docx</td>
</tr>
<tr>
<td>b1-Billing Basics.docx</td>
</tr>
<tr>
<td>b2-Billing Management.docx</td>
</tr>
<tr>
<td>b3-Billing Administration.docx</td>
</tr>
<tr>
<td>b4-Billing Development.docx</td>
</tr>
<tr>
<td>c0-CRM Overview.docx</td>
</tr>
<tr>
<td>c1-CRM Basics.docx</td>
</tr>
<tr>
<td>c2-CRM Management.docx</td>
</tr>
<tr>
<td>c3-CRM Administration.docx</td>
</tr>
<tr>
<td>c4-CRM Development.docx</td>
</tr>
</tbody>
</table>

**Doc-To-Help**

Would you like to view the help target that has just been built?

- [ ] Don't show me this again
Eyes on your goals.

Reuse It

Build deliverables from topics (files)
Home stretch!

1. Define reuse tags
2. Set tags on each file
3. Define build targets with tag filters
4. Automate builds
First, define all of the attributes (facets) that you will want to filter on.
Second, define easy-to-interpret values for filtering.
For each file, right-click and set Properties

Under **Condition**, set Attributes
Best use of Condition types

- To stage pre-release files, restrict by **Attribute**
- To limit to output type (HTML only), restrict by **Platform**
- For *custom* control (mix-n-match), restrict to one or more **Target**
Now, go define your “targets” (deliverables)

Future-proof and flexible: If you later add a ‘Reporting’ module, it will pick up `r3-Reporting_Admin.docx` by attributes
Examples of Targets

**Event Guide**
- e0-Events Intro
- e1-Events Basics
- e2-Events Manage
- e3-Events Admin

**Finance Guide (custom)**
- b2-Billing
- b3-Billing Admin
- c3-CRM Admin
- e3-Events Admin
- a2-Keyboard

**Bootcamp**

**Intro Course**
- *0-* (Newbie)
- *1-* (Lite)

**Staff Course**
- *2-* (Staff)
- *3-* (Admin)

**Developer**
- *0-* (Newbie)
- *4-* (Dev)
Final Goals

1. Single-sourcing
   - Across products (reuse)
   - Across outputs (HTML, PDF)

2. Automatic outputs
   - No-touch builds
   - No-touch delivery

3. Need a new deliverable?
   Just add a new Build Target!
Use **Build Scheduler** and/or batch files to deliver outputs.

Best: Run D2H on dedicated build machine for hourly jobs.

Tip: Schedule builds for noon and midnight when GA is near.
D2H with TFS Setup

D2H installed on test + build machines
D2H builds from TFS working files (get)
One project generates all targets across all products (= shared content library)
D2H project (non-content) files managed separately in source control
Content editable on any TFS machine
Word in TFS

- Doc-To-Help project (Access DB)
- Team Foundation Server (VS) Source Control
- Sandcastle API reference
- Non-D2H .DOT

- Project folder
- Working folder
- Build targets
- Build log files

Product A Helpsite, DOC downloads, doc, html, exe

{ links to HTMs } in DOCX
Thanks!

Questions?

Reaching me:
maryconnor@gmail.com
@maryfconnor
MADWORLD
The Premier Technical Communication and Content Strategy Conference

APRIL 2-5, 2017 | HARD ROCK HOTEL | SAN DIEGO, CALIFORNIA

Register by February 28 and Save Up to $250 on MadWorld 2017

WWW.MADCAPSOFTWARE.COM/EVENTS/MADWORLD
Credits

Special thanks to all the people who made and released these awesome resources for free:

◎ Presentation template by SlidesCarnival
◎ Photographs by Unsplash
Presentation design

This presentation uses the following typographies and colors:

- **Titles:** Nixie One
- **Body copy:** Varela Round

You can download the fonts on this page:
http://www.google.com/fonts/#UsePlace:use/Collection:Nixie+One|Varela+Round

Click on the “arrow button” that appears on the top right

- Yellow: #f8bb00
- Blue: #00acc3
- Green: #65bb48
- Orange: #ed4a00
- Aqua: #00d1c6
- Gray: #617a86
- Fucsia: #e8004c
- Lime: #bbcd00
- Light Gray: #a1becc