



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

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

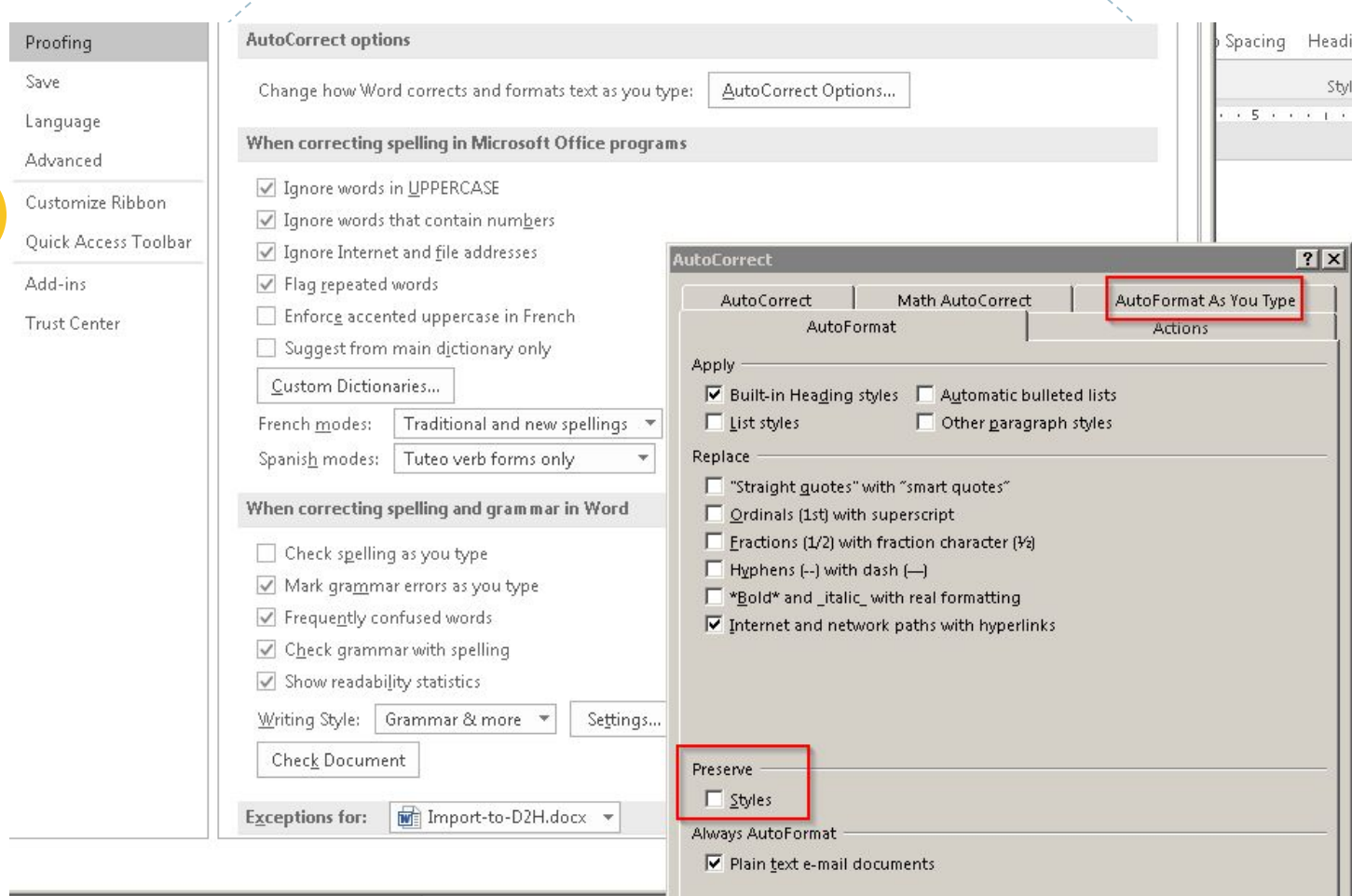
Team features: source files in TFS,
SharePoint; familiar editors, no markup

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:

The screenshot displays the Microsoft Word application window. The **View** tab is selected in the ribbon, with the **Developer** tab also highlighted. In the **Views** group, the **Outline** and **Draft** view buttons are highlighted with red boxes. In the **Show** group, the **Ruler**, **Gridlines**, and **Navigation Pane** checkboxes are highlighted with red boxes. The **Navigation** pane on the left shows a document structure with the following items:

- Swarm Concepts
 - Understanding Swarm Objects
 - Types of Data Objects** (selected)
 - Types of Container Objects
 - Understanding Unnamed Objects
 - Understanding Named Objects
 - Working with Swarm Objects
 - Naming Rules
 - Universal Resource Identif...
 - Content Cache
 - Data Protection
 - Understanding Feeds
 - Purpose of Feeds
 - Feed Plug-ins
 - Elastic Content Protection
 - Content Protection with R...
 - Content Protection with E...
 - Conversion between Content Prote...

The **Find and Replace** dialog box is open, showing the **Find** tab. The **Find what:** field contains the text `^p^w`, and the **Replace with:** field contains the text `^p`. The **More >>** button at the bottom left of the dialog is highlighted with a red box. A red speech bubble labeled **Style pane** points to the **Types of Data Objects** item in the Navigation pane. The main document area shows the title **Understanding Swarm Objects** and the subtitle **Types of Data Objects**, followed by the text: "All data objects used by Swarm are either unnamed or..." and a bullet point: "• **Unnamed objects** are assigned identifiers by Swa... and deleted using an unambiguous ID. Swarm is...".

Our Scope: Soup to Nuts!

1. Standardize it



2. Structure it



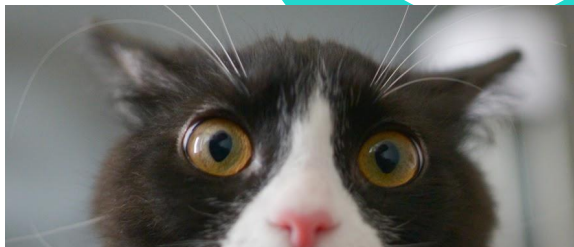
3. Reuse it



The image features two ginger cats in profile, facing each other. A large, solid blue number '1' is positioned between them. The background is white with various colorful geometric shapes: orange and yellow circles in the top left, a green circle in the top right, a large blue circle in the bottom right, and a green circle in the bottom left. Some of these circles have dashed outlines. The cats are looking upwards and to the right.

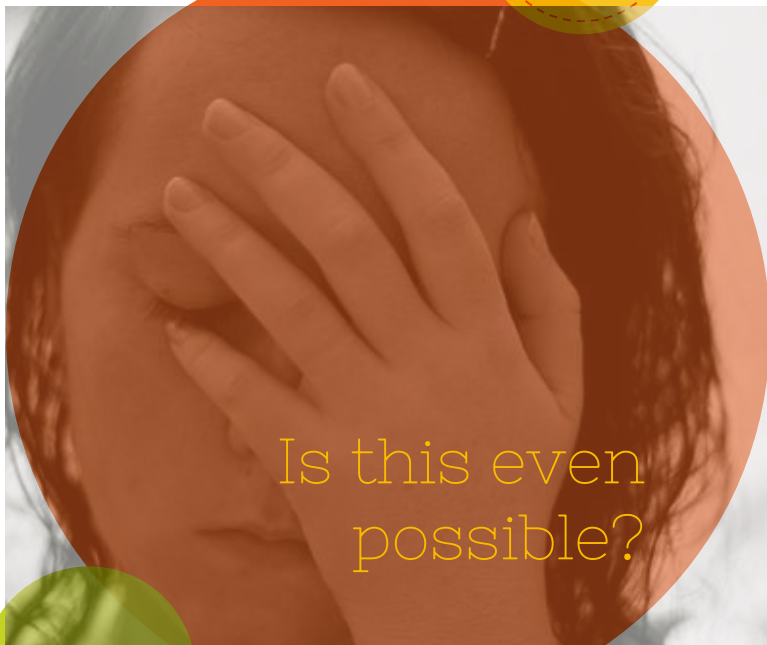
1

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



Is this even
possible?

1- UNIFY

Case Studies:

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

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



Migrating from a CCMS*

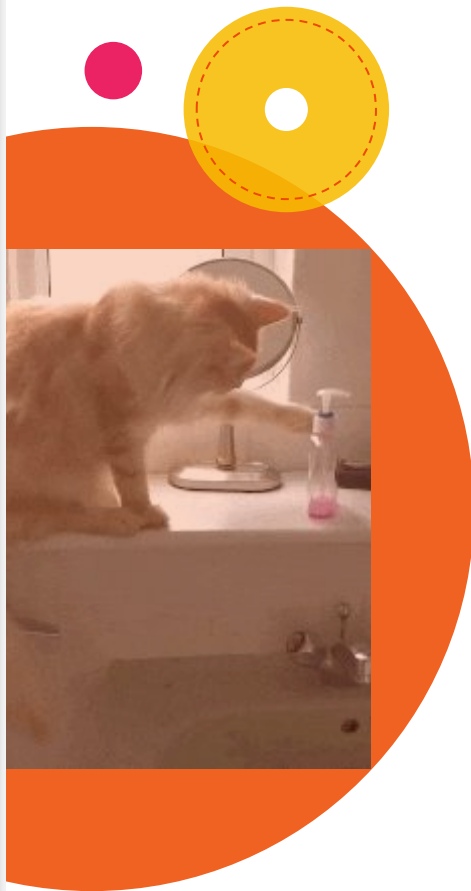
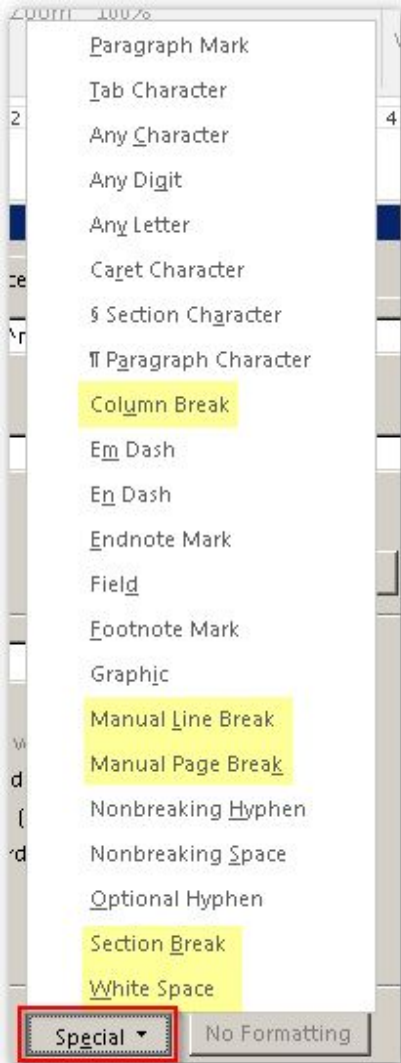
- ❑ Unraveled existing reuse so each “object” publishes just once
- ❑ Built new book to export all to a single, linked DOC
- ❑ Optimized templates for D2H import with Word macros to post-process: resizing graphics, substituting styles, deleted section breaks, etc.
- ❑ Automated migration so I could iterate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

*component content management system

Migrating from DocBook

- © Used **xmllint** 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

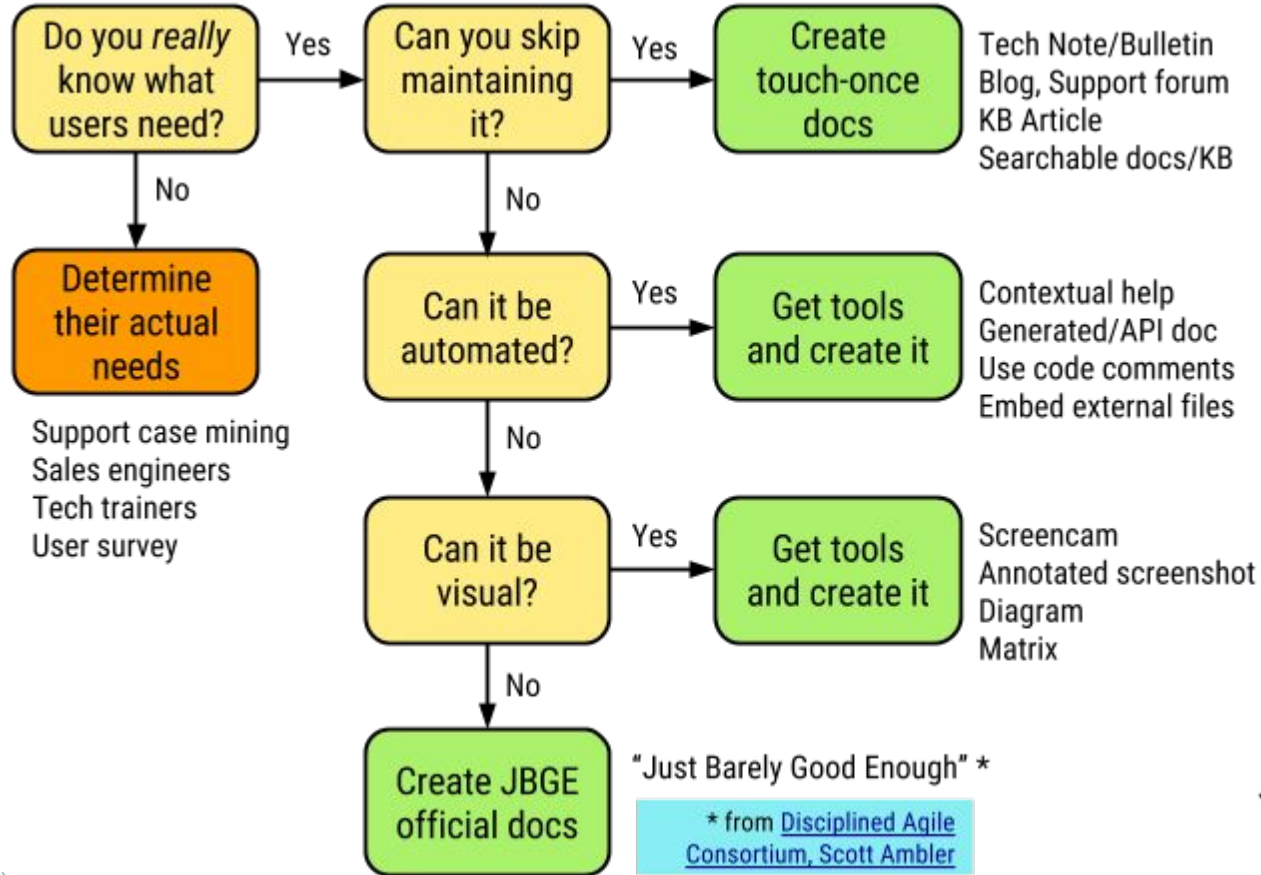
```
1 REM Project to export Docbook to Word, February 2015
2
3 REM Pandoc for doc conversion; see http://johnmacfarlane.net/pandoc/README.html
4
5 cd C:\svn\Swarm\Docbooks\CASTorAdvancedAdministrationGuide
6 xmllint --xinclude CASTorAdministrationGuide.xml --output merged.xml
7 pandoc -s -f docbook -t docx -o CASTorAdministrationGuide.docx merged.xml
8 pandoc -s -f docbook -t markdown -o CASTorAdministrationGuide.txt merged.xml
9
```

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

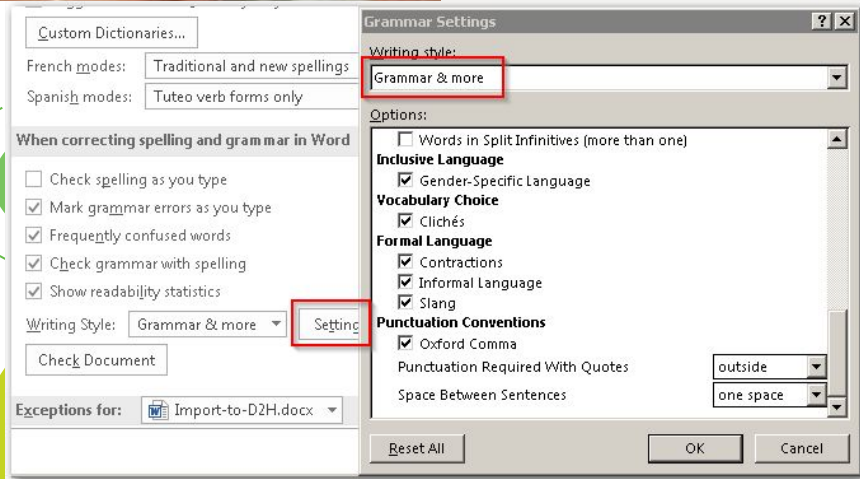
Survival: Do more with less.





...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)

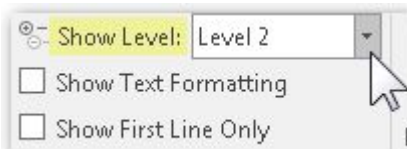


3- Standardize Headings

- ◎ 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
 - **Introduction**



Make Headings Orderly, Unique, and Standalone!



Heading 1	⊕ CASTor
Heading 1	⊖ Introduction
Heading 1	⊕ <u>Swarm Concepts</u>
Heading 1	⊕ <u>Planning and Setup</u>
Heading 2	⊕ <u>Installing the Cluster</u>
Heading 2	⊖ <u>Configuring the Cluster</u>
Heading 2	⊕ <u>Defining Admins and Users</u>
Heading 1	⊕ <u>Managing the Cluster</u>
Heading 2	⊕ <u>Advanced Administration</u>
Heading 1	⊕ <u>Development Essentials</u>
Heading 2	⊕ <u>API Reference</u>
Heading 2	⊕ <u>Advanced Development</u>
Heading 1	⊕ CloudScaler
Heading 1	⊖ Introduction
Heading 1	⊕ <u>Planning and Setup</u>
Heading 2	⊕ <u>Installing the Cluster</u>
Heading 2	⊖ <u>Configuring the Cluster</u>
Heading 2	⊕ <u>Defining Admins and Users</u>
Heading 1	⊕ <u>Managing the Cluster</u>
Heading 2	⊕ <u>Advanced Administration</u>
Heading 1	⊕ FileFly
Heading 1	⊖ Introduction
Heading 1	⊕ <u>Planning and Setup</u>
Heading 2	⊕ <u>Installing the Cluster</u>
Heading 2	⊖ <u>Configuring the Cluster</u>
Heading 2	⊕ <u>Defining Admins and Users</u>
Heading 1	⊕ <u>Managing the Cluster</u>
Heading 2	⊕ <u>Advanced Administration</u>

Fix h1 level,
then h2, ...

Heading 1	⊕ CASTor
Heading 1	⊖ CASTor Overview
Heading 1	⊕ <u>CASTor Concepts</u>
Heading 1	⊕ <u>CASTor Network and Hardware</u>
Heading 2	⊕ <u>CASTor Installation and Upgrade</u>
Heading 2	⊖ <u>CASTor Cluster Configuration</u>
Heading 2	⊕ <u>CASTor Access Control</u>
Heading 1	⊕ <u>CASTor Cluster Management</u>
Heading 2	⊕ <u>CASTor Advanced Administration</u>
Heading 1	⊕ <u>CASTor Development Essentials</u>
Heading 2	⊕ <u>CASTor API Reference</u>
Heading 2	⊕ <u>CASTor Advanced Development</u>
Heading 1	⊕ CloudScaler
Heading 1	⊖ CloudScaler Overview
Heading 2	⊖ CloudScaler Hardware
Heading 2	⊕ <u>CloudScaler Installation</u>
Heading 2	⊖ <u>CloudScaler Configuration</u>
Heading 1	⊕ <u>CloudScaler Access Control</u>
Heading 2	⊕ <u>CloudScaler Management</u>
Heading 2	⊕ <u>CloudScaler Advanced Administration</u>
Heading 1	⊕ FileFly
Heading 1	⊖ FileFly Overview
Heading 1	⊕ <u>FileFly Setup</u>
Heading 2	⊕ <u>FileFly Installation</u>
Heading 2	⊖ <u>FileFly Configuration</u>
Heading 2	⊕ <u>FileFly Access Control</u>
Heading 1	⊕ <u>FileFly Management</u>
Heading 2	⊕ <u>FileFly Advanced Administration</u>

2

Structure It For Topic-Base Authoring




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



* <https://www.linkedin.com/pulse/dita-topic-based-writing-flip-sides-same-coin-rob-gillespie>



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

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)

<http://ecolearnit.ifas.ufl.edu/concept.asp>



Encapsulation by RLO

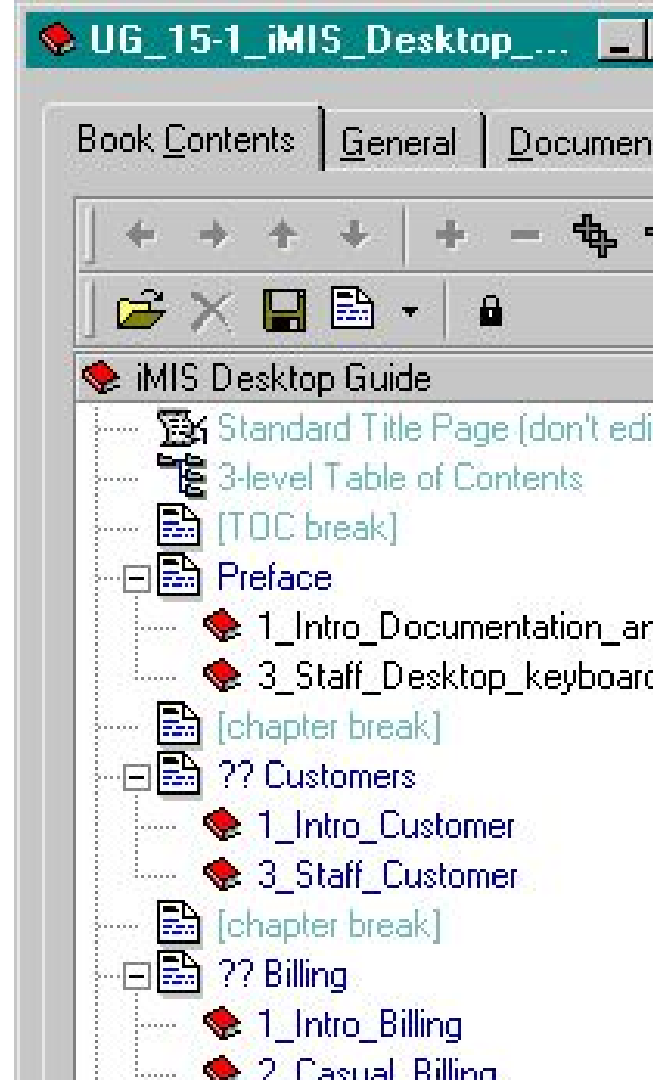
Approach (think, “*module-only*”)

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

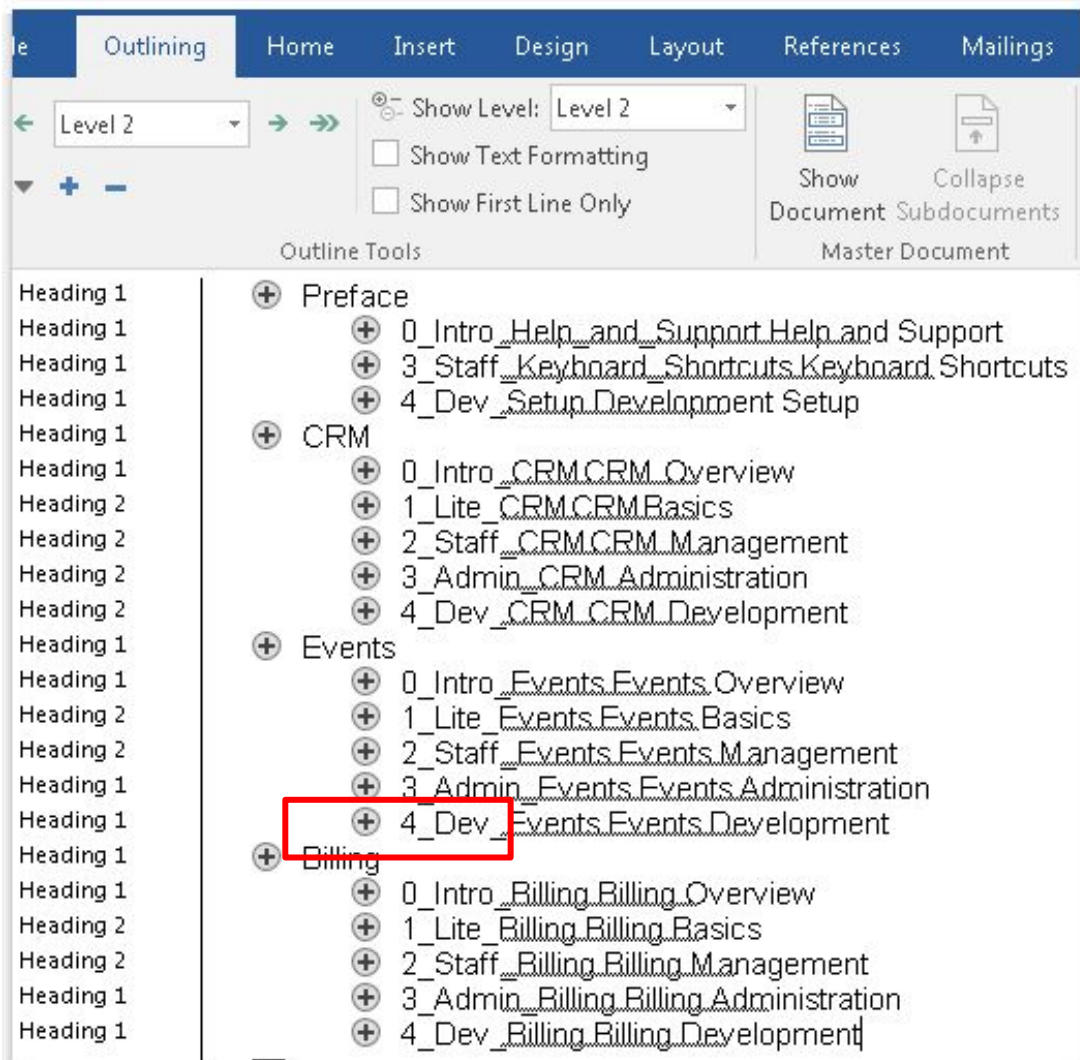
0 = all, prospect, mkt

1 = cheap license

2 = staff license

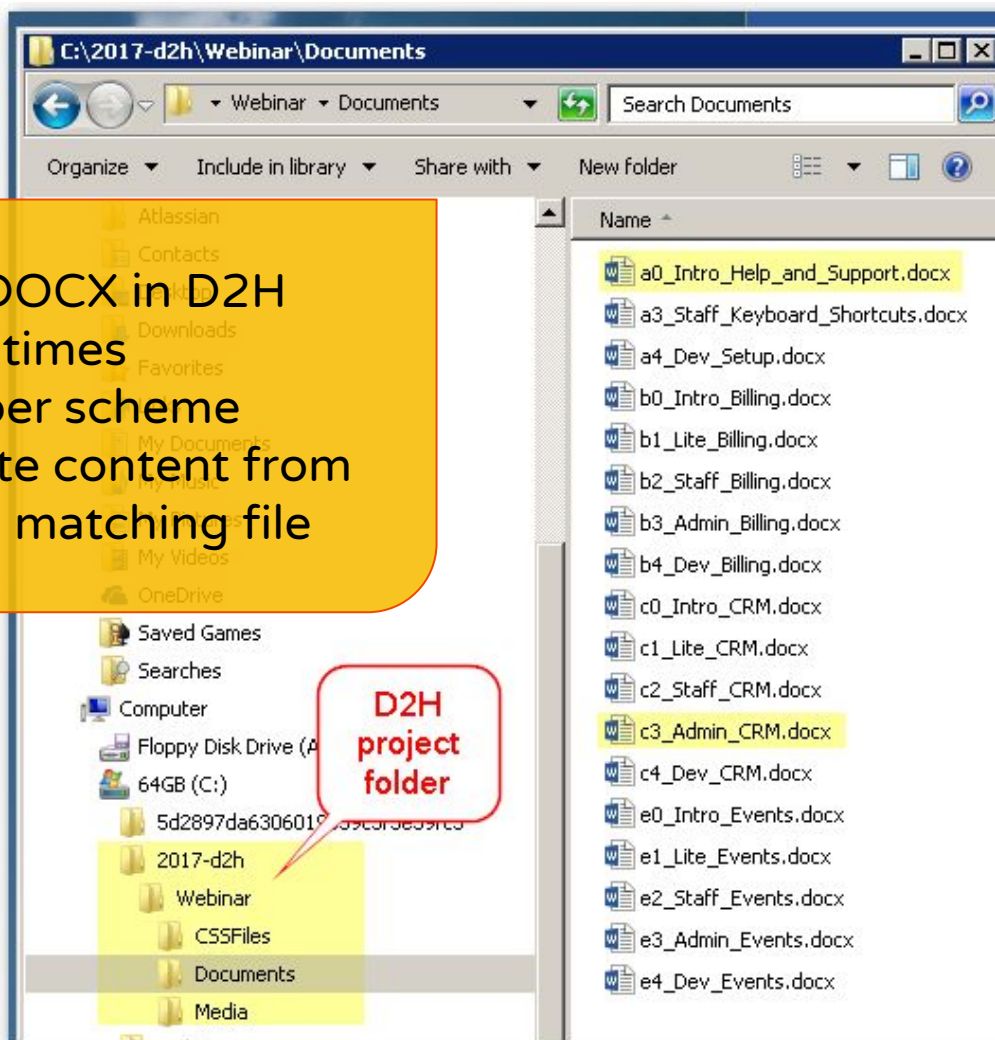
3 = admin access

4 = developer



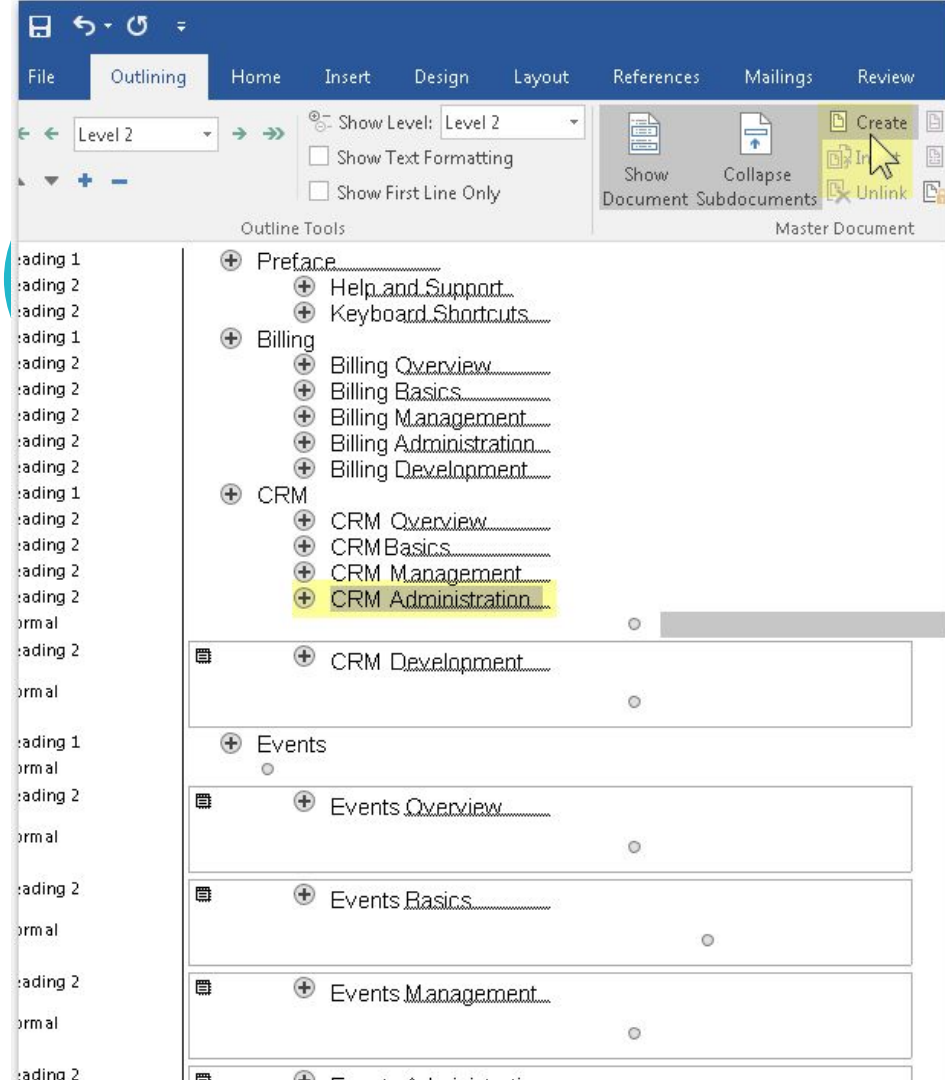
1. Create a DOCX in D2H
2. Clone it n times
3. Rename per scheme
4. Copy/paste content from Outline to matching file

**D2H
project
folder**



Shortcut!

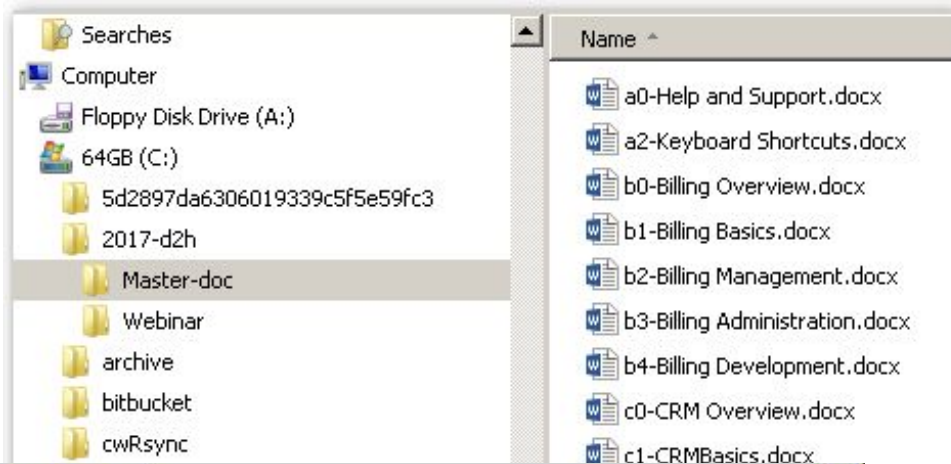
1. Make a copy
2. Attach D2H template
3. Use **Master Document** to quickly **Create** subdocuments



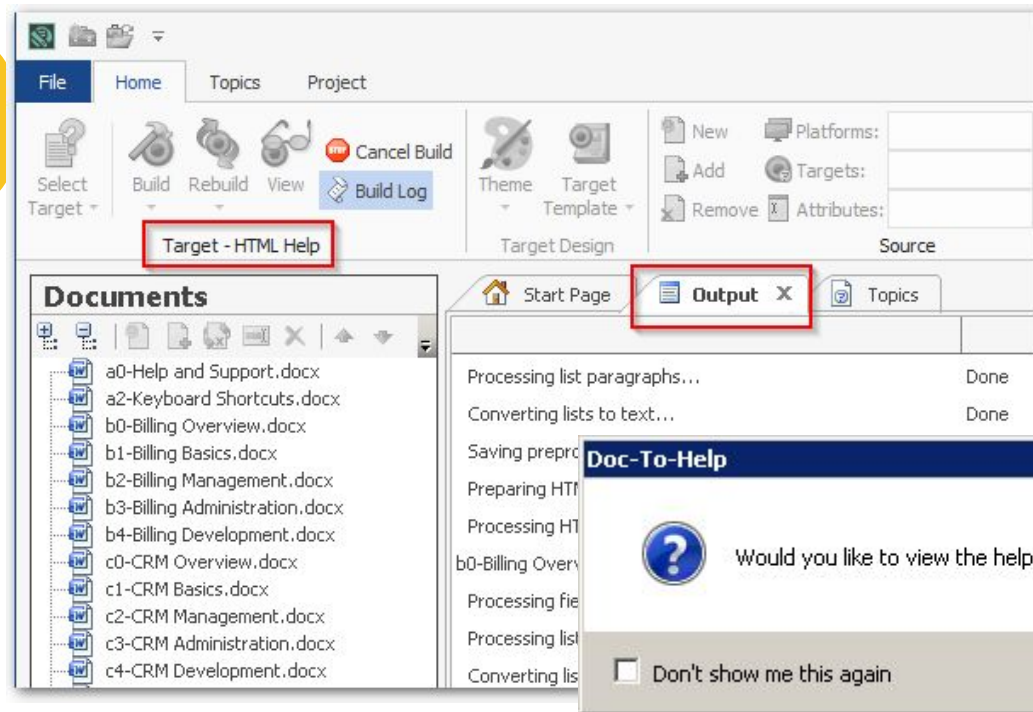
...Then have
D2H import
the entire
folder

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

Powerful flexibility!



You can build immediately!





3



Reuse It
Build deliverables from topics (files)

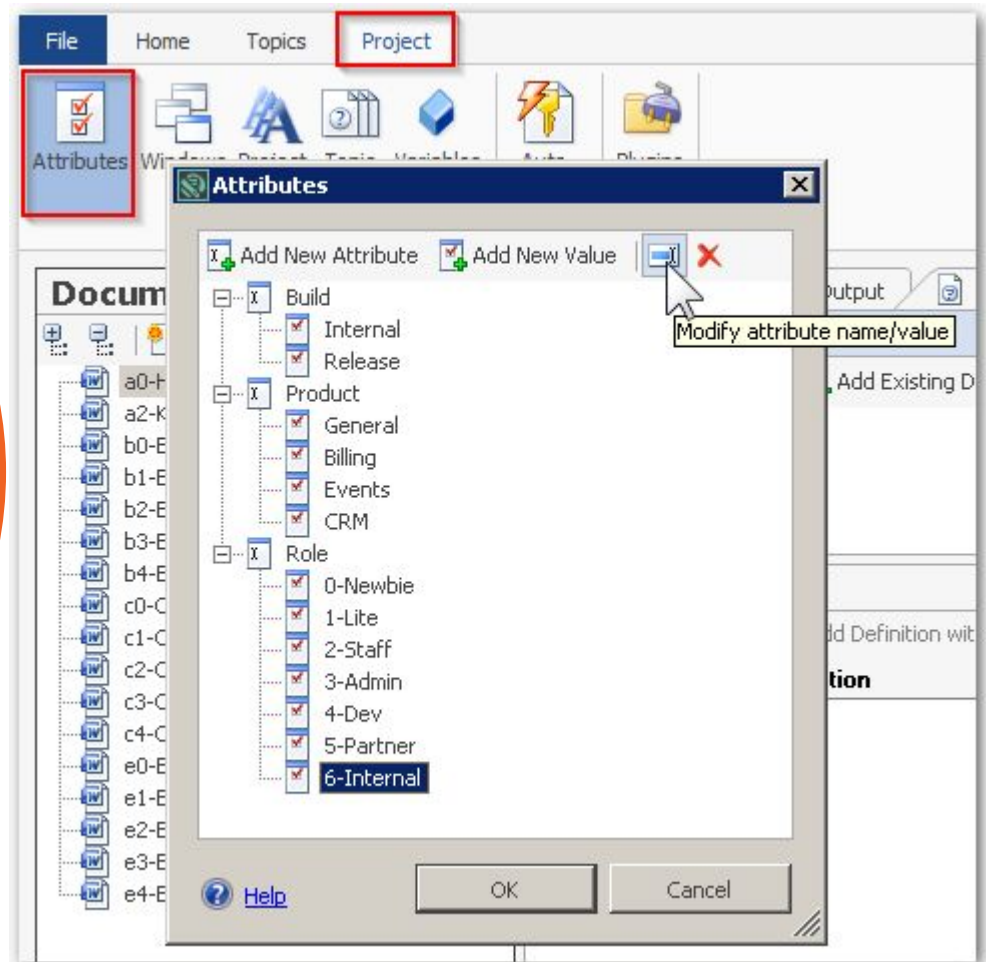
Home stretch!

1. Define reuse tags
2. Set tags on each file
3. Define build targets with tag filters
4. Automate builds



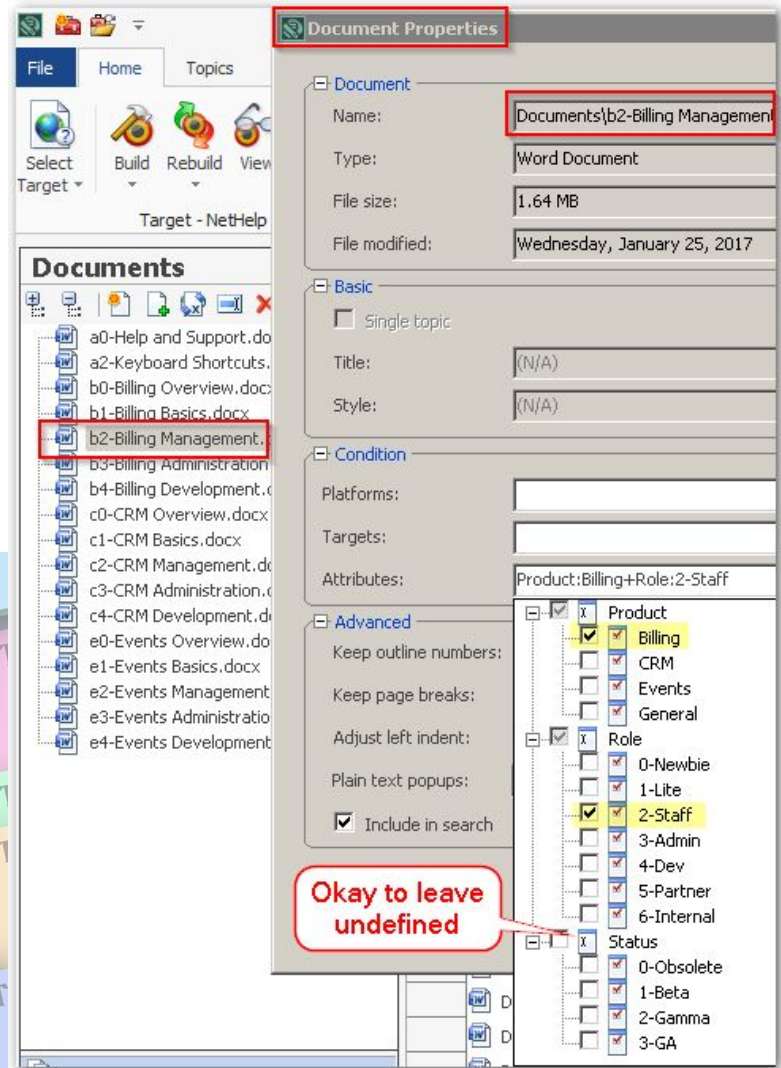
First, define all of the attributes (facets) that you will want to filter on

Second, define easy-to-interpret values for filtering



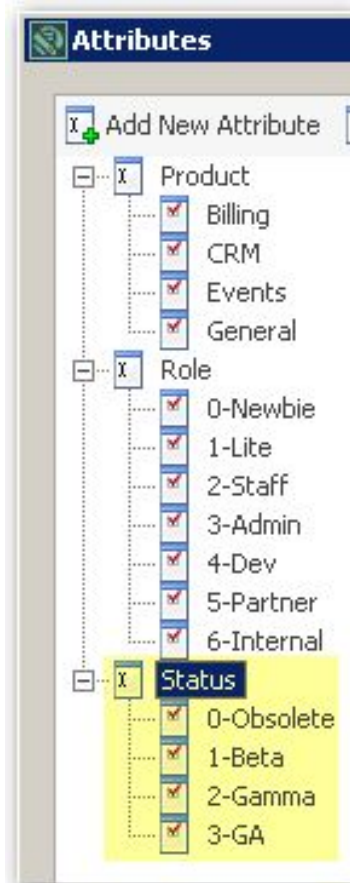
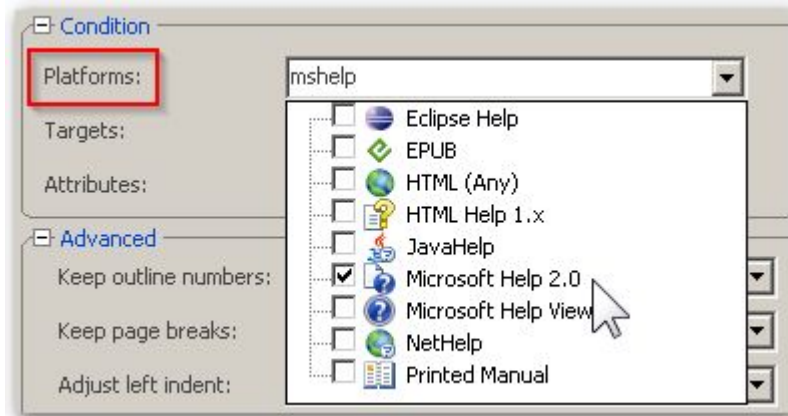
For *each file*,
right-click and set
Properties

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)

Basic

Name: _Acme_Admin_Coursebook

Base name: Acme_Administration

Style sheet: Print_nomargin.css

Template: C1H_PRNOMARGIN.DOT

Attributes: Role:3-Admin, Status:3-GA

Folder:

Attributes

Attribute	Value
Product	<input type="checkbox"/>
Role	<input checked="" type="checkbox"/>
Status	<input checked="" type="checkbox"/>

Role

Role	Value
0-Newbie	<input type="checkbox"/>
1-Lite	<input type="checkbox"/>
2-Staff	<input type="checkbox"/>
3-Admin	<input checked="" type="checkbox"/>
4-Dev	<input type="checkbox"/>
5-Partner	<input type="checkbox"/>
6-Internal	<input type="checkbox"/>

Status

Status	Value
0-Obsolete	<input type="checkbox"/>

Basic

Name: _CRM_Help

Base name:

Style sheet: NetHelp.css

Template: C1H_HTML.DOT

Theme: Responsive

Attributes: Product:CRM, Role:0-Newbie, Role:1-Lite, I

Folder:

Attributes

Attribute	Value
Product	<input checked="" type="checkbox"/>
Role	<input checked="" type="checkbox"/>
Status	<input checked="" type="checkbox"/>

Product

Product	Value
Billing	<input type="checkbox"/>
CRM	<input checked="" type="checkbox"/>
Events	<input type="checkbox"/>
General	<input type="checkbox"/>

Role

Role	Value
0-Newbie	<input checked="" type="checkbox"/>
1-Lite	<input checked="" type="checkbox"/>
2-Staff	<input type="checkbox"/>
3-Admin	<input type="checkbox"/>
4-Dev	<input type="checkbox"/>

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)



Set it
and
forget it

Final Goals

1. **Single-sourcing**

- Across products (reuse)
- Across outputs (HTML, PDF)

2. **Automatic outputs**

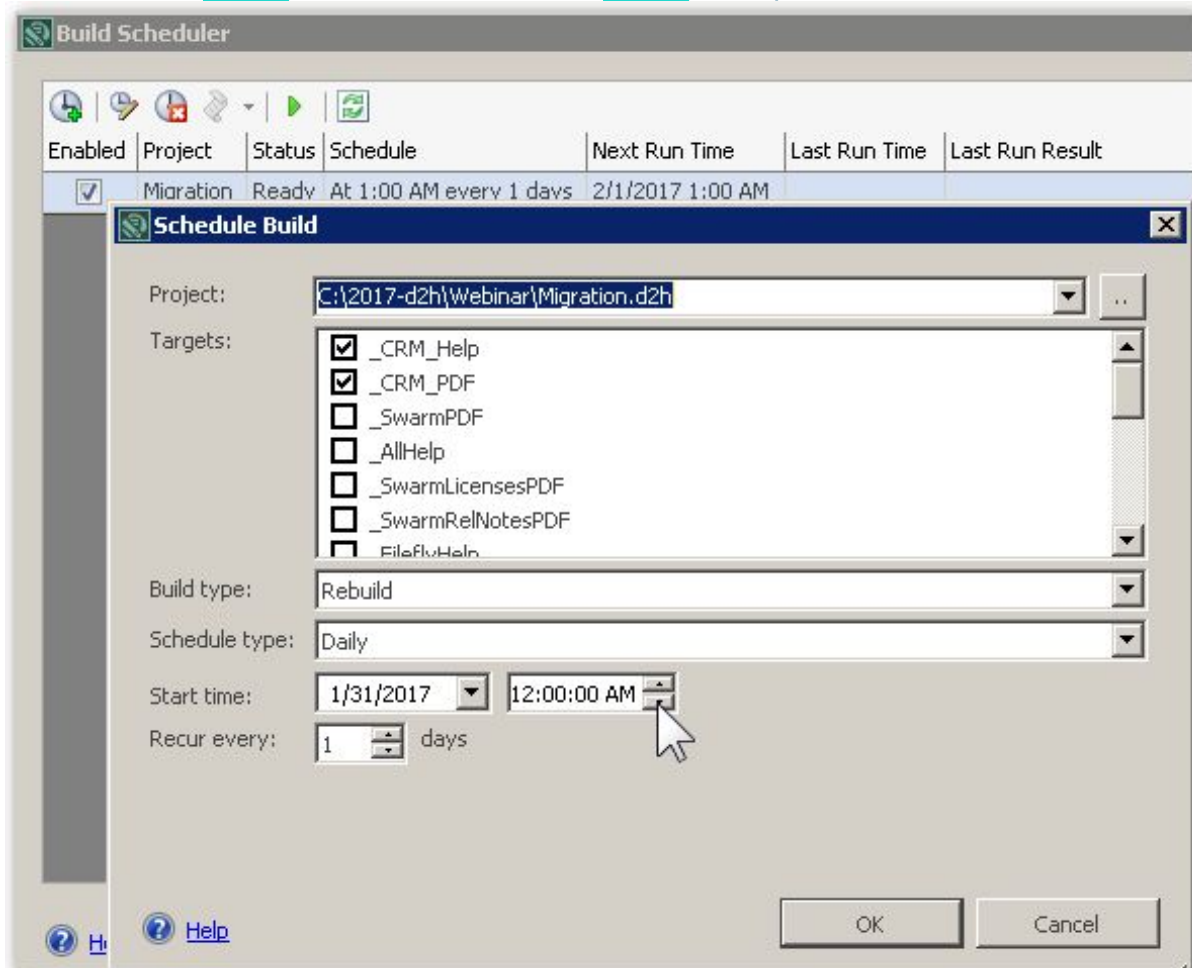
- No-touch builds
- No-touch delivery

3. **Need a new deliverable?** Just add a new Build Target!

Tip: Schedule builds for noon *and* midnight when GA is near

Use **Build Scheduler** and/or batch files to deliver outputs

Best: Run D2H on dedicated build machine for hourly



The background features several overlapping circles in teal, orange, yellow, and green. On the left side, there is a wooden toy box with blue sides and a red handle, filled with various wooden blocks and toys. The text is positioned on the right side of the slide.

D2H with TFS Setup

D2H installed on test + build machines

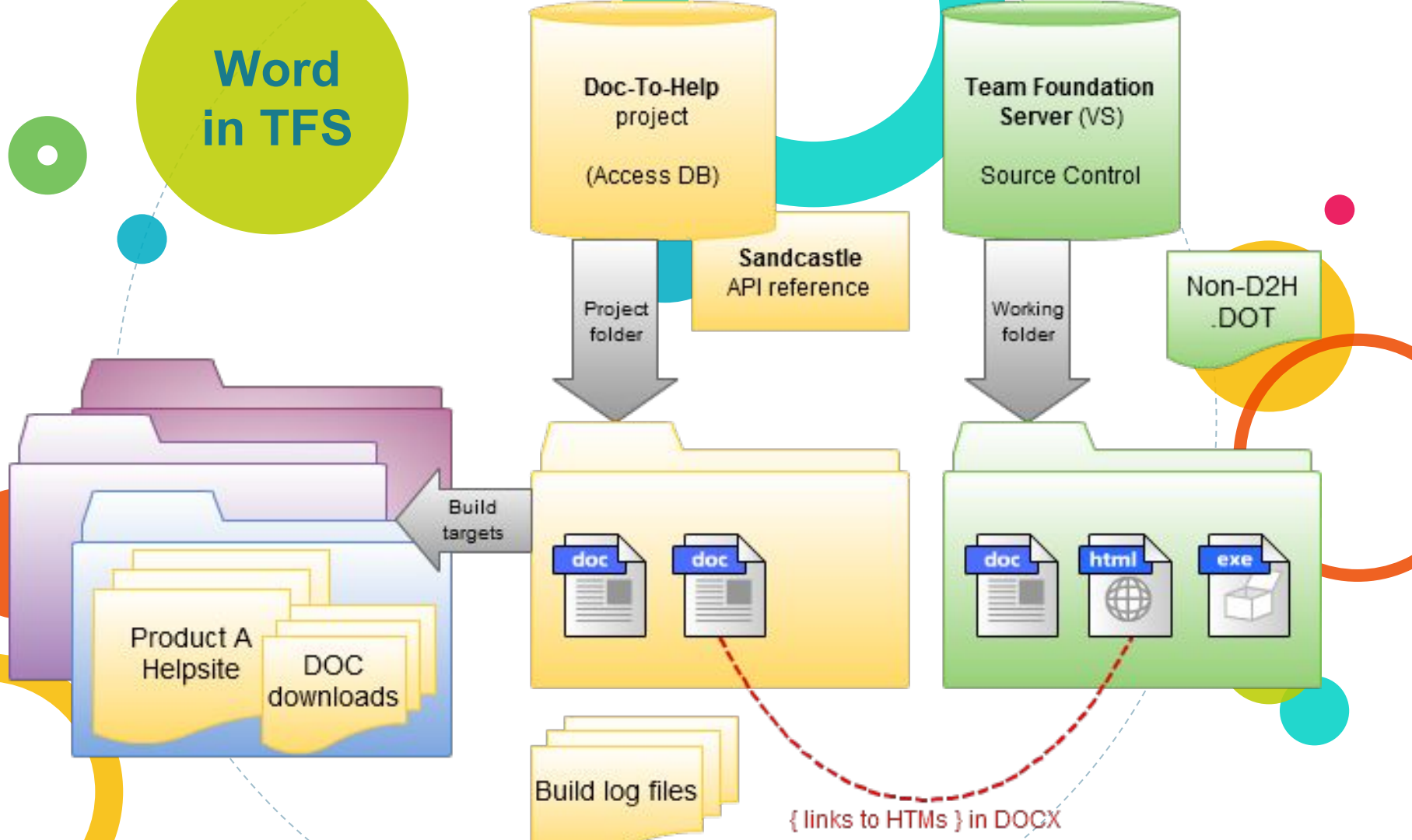
D2H builds from TFS working files (get)

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





Thanks!

Questions?



?

Reaching me:

maryconnor@gmail.com

@maryfconnor





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

Fuchsia #e8004c

Lime #bbcd00

Light Gray #a1becc