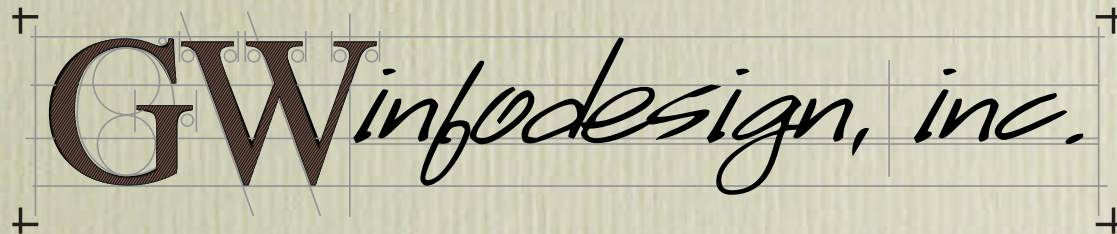


# Converting from FrameMaker® to Flare? We've Got Your Back

---

John F. Welle

July 24, 2013



# Who am I?



- Technical Writer for 18+ years;  
incorporated consultant 11+ years
- B.S. Cognitive Science;  
Master of Technical & Professional Writing
- Certified Flare Instructor & Developer
- FrameMaker user since version 4; print-focused and  
heavily involved in template design
- Flare user since version 1.0; started from scratch
- Learned HTML/CSS to manipulate Flare output



# Who are you?

- You have a bunch-o-doc to get into Flare
- Ideally, you:
  - ▶ Are not under the gun
  - ▶ Are not working on production doc
  - ▶ Are familiar with FrameMaker styles
  - ▶ Are able/willing to spend [considerable] time planning and experimenting before diving in
  - ▶ Know a bit of HTML/CSS (or are not shy about learning — see Pam Coca's recent webinar)
  - ▶ Do not seek the magic bullet











# This is a case study, sorta...

- I'm coming at this from a large-project perspective, but many concepts apply to any conversion project
  - ▶ Composite case study from multiple clients
  - ▶ The largest project involved converting 57 Frame docs (~3,000+ pages) in 10 weeks (400 hours, on bid)
- Will discuss some basic process strategies
- Will show some issues you may encounter and some ways of dealing with them
  - ▶ Pre-import cleanup on the FrameMaker side
  - ▶ Post-import cleanup on the XHTML side



# Where we're going

- |  |                      |   |                      |
|--|----------------------|---|----------------------|
|  phase<br><b>1</b>   | Evaluate and analyze |  phase<br><b>5</b>   | Document the process |
|  phase<br><b>2</b>   | Start small          |  phase<br><b>6</b>   | Import to Flare      |
|  phase<br><b>3</b>  | Tame your styles     |  phase<br><b>7</b>  | XHTML cleanup        |
|  phase<br><b>4</b> | FrameMaker cleanup   |  phase<br><b>8</b> | Think big            |



# Is it worth converting at all?

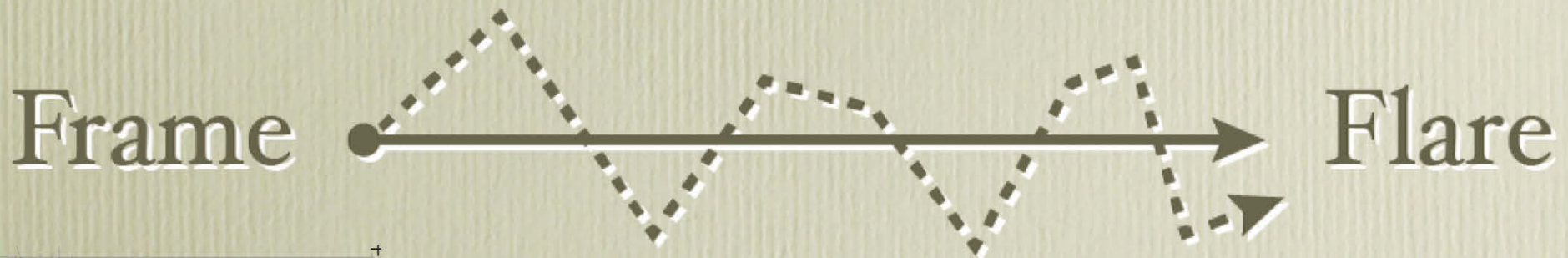
- Analyze the payoff: time spent vs. gain
  - ▶ For a small number of short docs, it may be easier to just cut & paste or export to text and format later
  - ▶ For a few longer documents/books, could use Flare import mainly as a mechanism to split content into topics, and massage the content later
  - ▶ For truly large projects, the time you spend working out a detailed conversion process can pay off





# Converting? Migrating? Both?

- Is the goal simply to switch tools and reproduce your FrameMaker output via Flare, or is there also major rewriting/restructuring needed?
  - ▶ Example: stable product, wanted richer help & CSH without chop-my-book-up-into-chunks mechanics
  - ▶ Example: two products merged under a new UI — focused on templates, not conversion
- Depending upon your goals, likely not a straight line

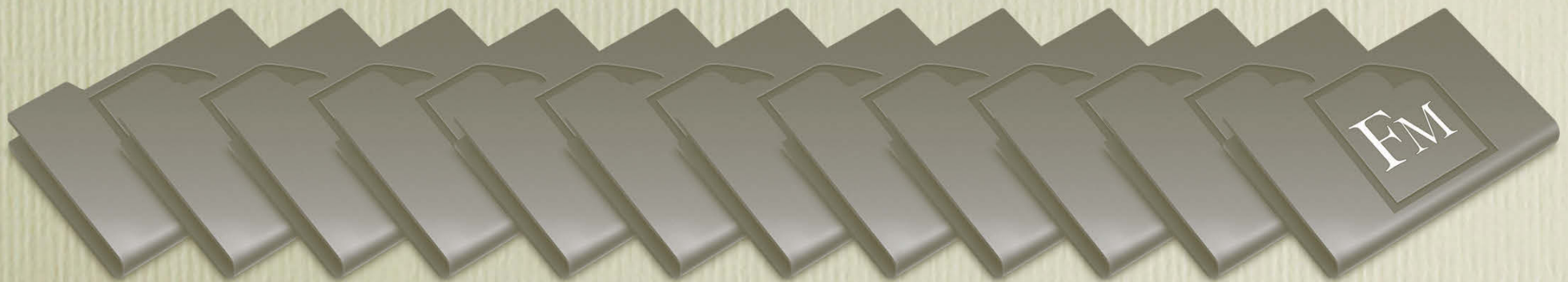






# See what you're dealing with

- Review a variety of documents — don't just take one as a sample (especially if it's not your content)
- Goal: Analyze & evaluate so you can make decisions on what to implement and what to let go of
- What styles are [actively] used? Are there Overrides?
- How consistent is the document structure?







# See what you're dealing with

- Are there any layout-dependent design elements?
  - ▶ Sideheads? Sidebars? (They 'can' be done in Flare, but only with an absolutely consistent structure)
- How are Notes/Tips implemented?

- ▶ As paragraphs?
- ▶ As tables?



## NOTE

At the end, include a Closing paragraph.

This inserts all the information, and I always be the last



*These buttons turn blue when you hover the mouse over them.*

- A bullet.
- Another bullet

- Is there conditional text?
  - ▶ Large sections or small chunks? How much?
  - ▶ How many conditions? Are they still relevant?





# See what you're dealing with

- What types of tables?

- ▶ How many formats?

- ▶ How complex? For data? For design?

- ▶ Do they have proper heading rows?

- How are numbers/bullets implemented?

- ▶ How many types? Are headings numbered?

- How are cross-references used?

- ▶ Only to headings, or to figure/table captions?

- ▶ References to specific list/step numbers?

Id el facilisse vulla nisim ornare lobor, aliquet consequipis lobortio quis ipisl  
dolortis. Ullaor scilis aciliqu vercilisit ese eriure dolessed

Id el facilisse vulla nisim ornare lobor, aliquet consequipis lobortio quis ipisl  
dolortis. Ullaor scilis aciliqu vercilisit ese eriure dolessed





# See what you're dealing with

- How are graphics used?
  - ▶ Are they embedded or referenced?
  - ▶ Which graphic formats?
  - ▶ Are there multiple objects inside anchored frames?
  - ▶ Where are frames anchored?
  - ▶ Any inline graphics?
- Anything else of note?

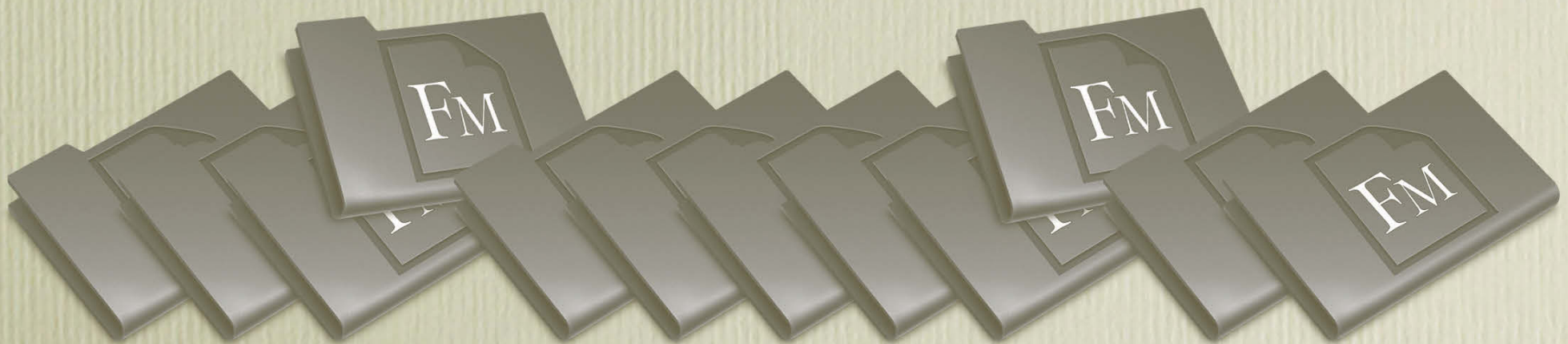




phase  
2

# Start small

- Select 2-3 representative documents
- Build a composite ‘sampler’ document if needed
- The goal is to capture a complete cross-section of the formats, structures, conventions, and quirks in the files you’ll be working with so you can test how you might either implement them or purge them

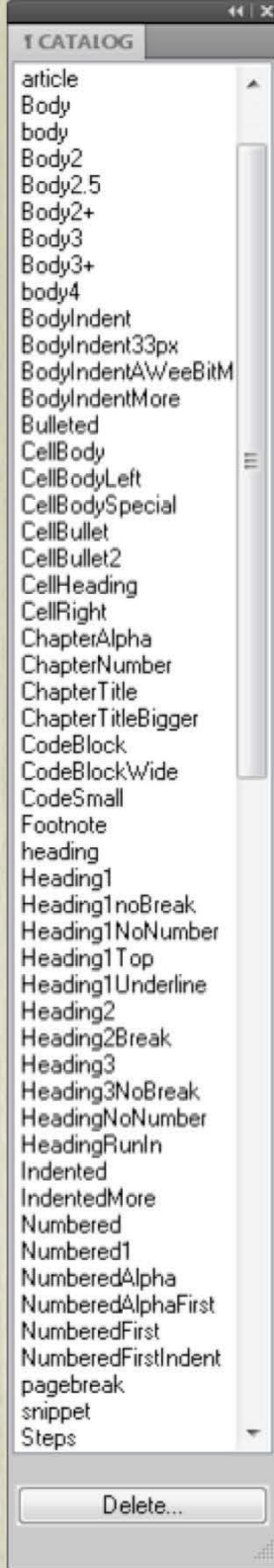




phase  
3

# Think 'clean and simple'

- Legacy FrameMaker documents accumulate cruft
  - ▶ Extra styles!
    - (paraIndent, paraIndentMore, paraIndentJustABitMore, paraIndentJustALittleWeeBitMore...)
  - ▶ Lots of overrides!
  - ▶ Extra master pages!
  - ▶ Extra reference pages!



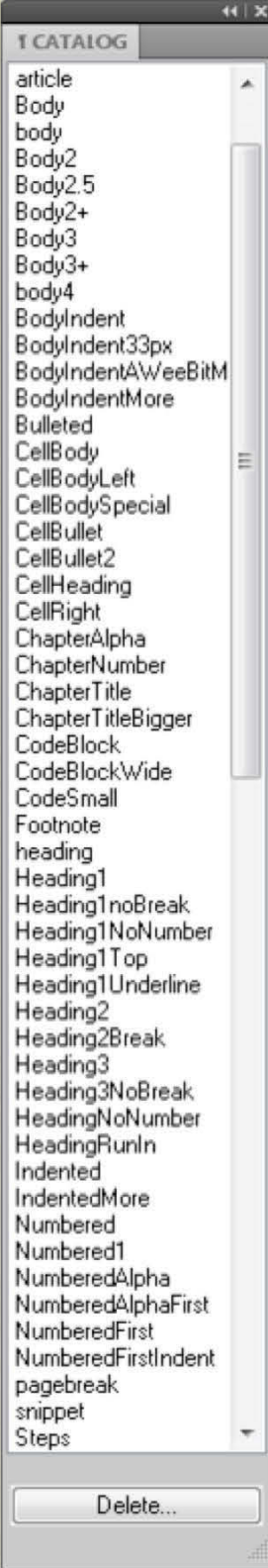


phase  
3

# Target the Cruft

- Wikipedia: “Cruft is jargon for anything that is left over, redundant and getting in the way. It is used particularly for superseded and unemployed technical and electronic hardware and useless, superfluous or dysfunctional elements in computer software.”

*“Whoa... hold on, John — Flare will ignore unused styles, and I can just delete the other junk after importing”*





phase  
3

# So is cleanup worthwhile?

- As I frequently work in code view, I am admittedly distracted by messy code ('code gremlins')
- It doesn't hurt to do a quick & dirty import of your book into Flare and see what happens
- If you're happy with the *visible* results then fine; if not then you have more work to do
- *So many ways* you could tinker...  
need to figure where to draw the line





phase  
3

# Decide what you ~~want~~ need

- Make a complete list of FrameMaker styles in use
- Take this as an opportunity to start fresh and trim down your style catalog
- Build a spreadsheet:
  - ▶ Make a complete list of FrameMaker styles in use
  - ▶ Make a list of Flare styles you need — start with the basics and expand from there as needed
- Don't hang on to old names for the sake of nostalgia
  - ▶ Preserve style names if in doubt; easy fix later



phase  
3

# Develop a style map

<i>Frame Style</i>	<i>Flare Style</i>	<i>Why?</i>
Body, TableBody	p	standard HTML; don't necessarily need special TableX formatting
Heading2Break, Heading2NoBreak, ...	h2	standard HTML; handle breaks with CSS
Code	p.code, span.code	likely want paragraph & character styles
stepFirst/Next, stepIndentFirst/Next, alphaFirst/Next, alphaIndentFirst/Next, tableStepFirst/Next,	p.numbered, p.alpha	Simplify! Document structure may not need First/Next styles





# Back up! ...*somewhere*

- *Always* work on a copy; keep an original somewhere



\*Dropbox is more sync than backup



- Duplicate your project folders before each major change so you can step back a notch or two if needed



**MyProject-original**



**MyProject-copy**



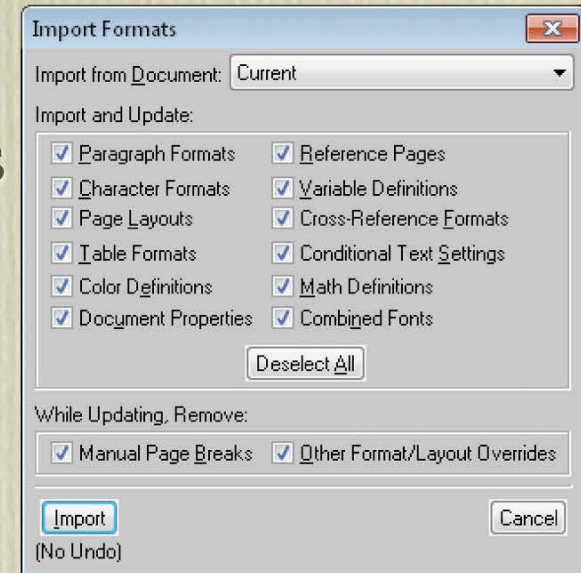
**MyProject-copy2**



# phase 4

## Purging FrameMaker cruft

- FrameMaker's built-in functions are a start
  - ▶ Easy to blow away style overrides by importing a document's own formats and clearing overrides
  - ▶ More tedious to manually delete styles from all of your documents and re-import from a clean template
  - ▶ May still have a lot of re-tagging to do
- But...imagine doing that on dozens of files — there are far more efficient ways of working





phase  
4

# Cue the third-party tools

- Silicon Prairie Software ([siliconprairiesoftware.com](http://siliconprairiesoftware.com))
  - ▶ Tools to list paragraph/character/table formats in use, delete unused formats, rename formats, remove overrides, and more
  - ▶ Also tools for variables, cross-references, etc.
  - ▶ Inexpensive, \$10/seat
- [Demo]



# And, for seriously heavy lifting...

- FrameScript ([framescript.com](http://framescript.com))
  - ▶ Very powerful — ideal for tasks that are much more complicated, yet still repetitive
  - ▶ ~~\$150~~/seat (*v6 on sale in July*), so may not be an option for those migrating unless there's really a lot to do
  - ▶ Rick Quatro ([rick@frameexpert.com](mailto:rick@frameexpert.com)) is master of all things FrameScript; he has a broad resource library
- If you have FrameScript, the [i]-frame tools are very useful, but in the \$hundreds ([i-frame.itl.info/en.html](http://i-frame.itl.info/en.html))
- [Demo]



phase  
5

# Before you import...



- Go back and update your style map/comments
- Document your process
  - ▶ Easy to forget the sequence of steps or the logic behind them (fwiw, my memory is also lousy...)
  - ▶ Essential if multiple people will be pitching in at the mass-conversion stage
- Make note of any other quirks you've found
- Make another backup
- Re-Caffeinate





# Document your process

▼ 2 Pare down existing styles	
▼ 2.1 Paragraphs	
• 2.1.1 Select book	
• 2.1.2 Run Tools > Paragraph Tools > List Paragraph Usage	get an idea of what's in there
• 2.1.3 Run !_Control_Script3 FrameScript	Strips content out of Code/Example tables, and deletes all reference pages (that's key here at the moment)
• 2.1.4 Run Tools > Paragraph Tools > List Paragraph Usage	List should be significantly shorter
• 2.1.5 Run Tools > Paragraph Tools > Delete Unused Paragraph Tags	
• 2.1.6 Run Tools > Paragraph Tools > List Paragraph Usage	should just be the top list now
▼ 2.2 Characters	
• 2.2.1 Select Book	
• 2.2.2 Run Tools > Character Tools > List Character Usage	If list is short and neat (Bold, Code, Normal, Relief), don't need to run character scripts
• 2.2.3 Run Tools > Character Tools > Delete Unused Paragraph Tags	
• 2.2.4 Run Tools > Character Tools > List Character Usage	Should be down to short list now
▼ 2.3 Paragraphs 2	
▼ 2.3.1 Open /Projects/Project_Templates/FrameMaker/Conversion_Templates/Style_Imports.fm	
• Select all files in book (not the book itself)	
• Do File > Import > Formats (Ifio)	
• Set Import from Document to Style_Imports.fm	
• Select only Paragraph formats from the top section; select both boxes in the bottom section	
• Click Import; click OK to dismiss the warning.	
• 2.3.2 Open /Projects/Project_Templates/FrameMaker/Conversion_Templates/ParaTagConversions.fm	
• 2.3.3 Run Tools > Paragraph Tools > Convert Paragraphs	
• 2.3.4 Run Tools > Paragraph Tools > Delete Unused Paragraph Tags	
• 2.3.5 Run Tools > Paragraph Tools > List Paragraph Usage	should just be the top list now
▼ 2.4 Characters 2	
• 2.4.1 look at character overrides; alternate between List Character Usage, Delete Unused Character Tags, Report: Format Overrides, and Convert Character Formattings to Tags	
▼ 3 Conditional text	
▼ 3.1 Run i-tools > Quality Pack > Report: Condition Formats	
• 3.1.1 if none used, run Quality Pack > Remove Conditions	



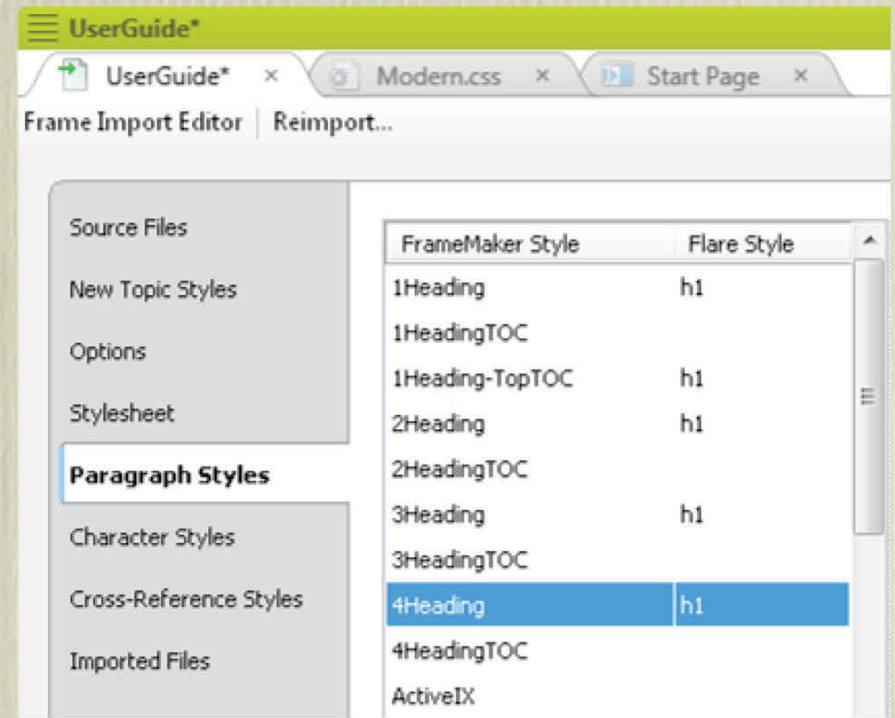
# phase 6

## Import to Flare (finally...)



- Remind yourself that this is a massively iterative process — import, examine, tweak, repeat
- There are oodles of import parameters to tweak
- You can create multiple import files (.flimp), each applied to different Frame files with different settings
- Avoid using subheadings within topics if you can, but don't map all of your heading levels to h1 yet (that's easy to fix later)\*


\* Sorry, that's a whole session of its own





# phase 6

## After you import...

- Pause for a moment of joyful reflection — you're now dealing *only* with plain text files 
  - ▶ Heck, you can edit them in Notepad if you want!
  - ▶ But, consider a powerful text editor that can do search & replace recursively across multiple directories at once via regex/grep\*
  - ▶ I use TextWrangler (Mac) or TextEdit (Windows); there are others such as BBEdit, Notepad++, etc.
- Get at least one license for MadCap Analyzer

\* More on regex/grep in a minute, but do note that Flare 9 does support this



# XHTML/CSS cleanup

- Code gremlins — entirely harmless, but annoying:
  - ▶ `<b style="font-weight:normal;">...</b>`
  - ▶ `<i> </i>`
  - ▶ `<span>...</span>`
  - ▶ `<![CDATA[ ]]>`
- Okay, but not so great:
  - ▶ `<b style="font-style:italic;">...</b>`
- Definitely not a good thing:
  - ▶ `<p style="...">...</p>`





# Batch search & replace

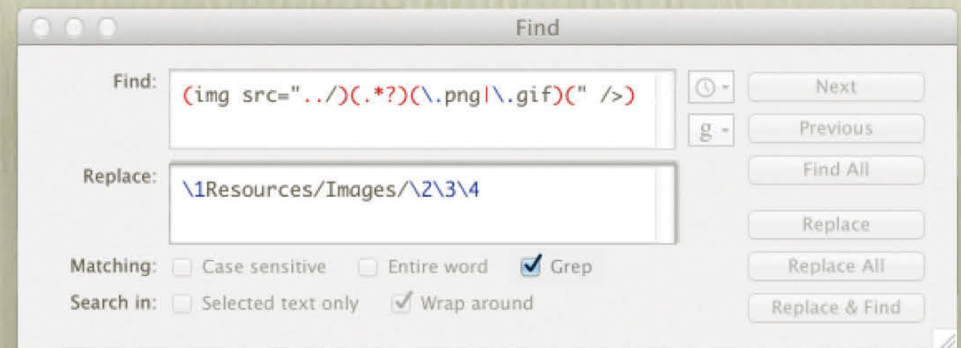
- Make a list of gremlins
- Advanced Find and Replace (abacre.com)
- Boom. Gone.

```
<![CDATA[ ]]>
<i></i>
<b></b>
<i> </i>
<b> </b>
<u></u>
<u> </u>
<i><![CDATA[ ]]></i>
<b><![CDATA[ ]]></b>
<span class="F__Times_New_Roman__Fs__12_00p" style="font-style: normal;"></span>
<span class="code"></span>
<span class="F__Times_New_Roman__Not_Bold"></span>
<span class="F__Times_New_Roman__Fs__12_00p__C__"></span>
<span class="F__Times_New_Roman__Fs__12_00p__C__L__Amer__English"></span>
<span class="F__Times_New_Roman__Not_Bold__Fs__12_00p"></span>
<span class="F__Times_New_Roman__Not_Bold__Not_Italic"> <![CDATA[ ]]></span>
<span class="F__Times_New_Roman__Not_Bold__Not_Italic"></span>
<span class="F__Times_New_Roman__Not_Bold__Not_Italic__C__"> <![CDATA[ ]]></span>
<span class="F__Times_New_Roman__Fs__12_00p__C__L__Brit__English"></span>
<span class="F__Times_New_Roman"> </span>
<span class="F__Times_New_Roman"></span>
<span class="F__Times_New_Roman__Fs__12_00p"></span>
<span class="F__Times_New_Roman__Fs__12_00p__C__L__Brit__English"></span>
<span class="F__Times_New_Roman__Not_Bold__Fs__12_00p__C__L__Brit__English"></span>
<span class="F__Times_New_Roman__Fs__12_00p"> <![CDATA[ ]]></span>
```



# Regular expressions and grep

- Search and replace on steroids
  - ▶ Backreferences; like search & replace with variables
- Not the most intuitive, but spending a few hours learning regex/grep can save many times that in repetitive drudgery
- Tools such as RegexBuddy/RegexMagic help develop complex search & replace expressions ([just-great-software.com](http://just-great-software.com))
- [Demo]





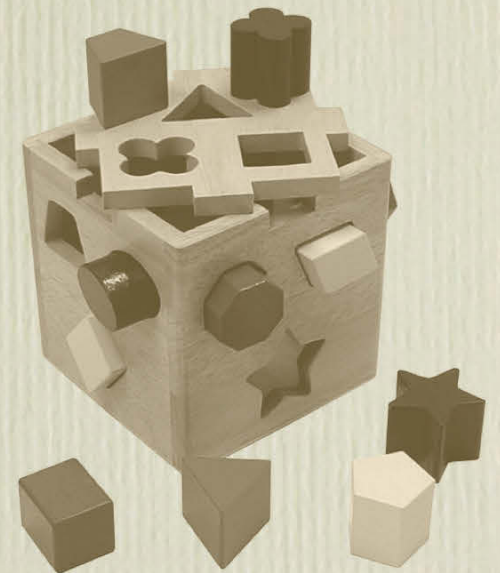
phase  
8

# Consider the big picture

- Make your life easier by organizing your topic and graphics files


Example: after importing, you have many dozens of graphics in a flat directory with your topic files:

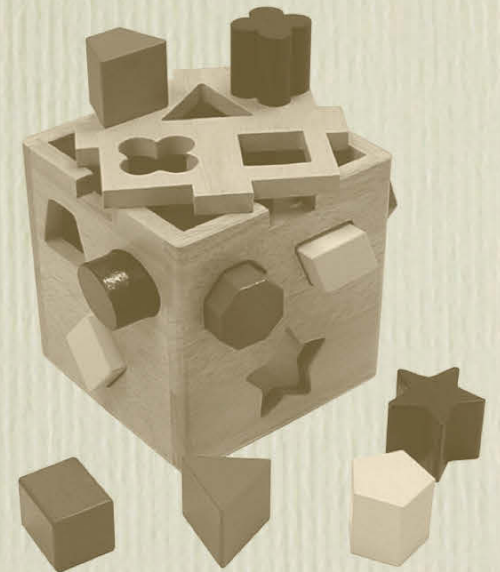
- ▶ From the Content Explorer, select all images and drag into /Content/Resources/Images
- ▶ Or, move them via Windows Explorer and relink them via search/replace [Demo]





# Consider the big picture

- File naming conventions are helpful
  - ▶ can be worked out in a later phase if necessary
  - ▶ Filename markers can be embedded in Frame documents before import, but it's a time sink
  - ▶ If you use icons inline for  buttons, etc., an img.icn class could drop them a tad below baseline and add a little extra padding...
- Consider your project architecture
  - ▶ How many projects might you need?
  - ▶ How might you use a Master Project?





# So what now?

- Update your process documentation
- Grab a few more documents and run them through your process to see if you get the same outcome
- Lather, rinse, repeat until you're comfortable with the results
- Grab a friend or unwitting co-worker and run the remaining documents through the conversion process





# To sum up...

phase  
**I** Evaluate and analyze

phase  
**2** Start small

phase  
**3** Tame your styles

phase  
**4** FrameMaker cleanup

phase  
**5** Document the process

phase  
**6** Import to Flare

phase  
**7** XHTML cleanup

phase  
**8** Think big



# MADWORLD

**ATTEND. LEARN. ADVANCE.**  
HARD ROCK HOTEL | SAN DIEGO, CA

**APRIL 13-15, 2014**

[www.MadWorldConference.com](http://www.MadWorldConference.com)





**\$100 OFF**



**madskills TRAINING**

**Thanks for attending today's webinar!**

As a webinar attendee, receive \$100 off any MadCap Training Course:  
**Discount Code MSTRN100\***

For available training courses and to receive your discount, contact:

**[sales@MadCapSoftware.com](mailto:sales@MadCapSoftware.com)**

**+1 858.320.0387 opt. 1**

**\*Valid for any current or future training course reserved by August 15, 2013**



# Questions?

- “anyone.....anyone.....”





# Thank you

[jwelle@gwinfodesign.com](mailto:jwelle@gwinfodesign.com)

