Single-Sourcing Complicated Content for Multiple Output Types: Mobile, HTML5, and PDF

Jennifer Hess, Principal Technical Writer
Trade Matching and Clearing Services Learning

15 June 2016
• Why go to multiple outputs and single-source content?
• Descriptions of the outputs.
• Setting up the outputs before generating them.
Per comScore\(^1\), in early 2014 the number of people accessing the internet with a mobile device exceeded desktop users.

Now, it’s not a question of mobile vs. desktop – it’s all of the above.

Integrate your content for all types of users to:

- Increase accessibility (↑ happiness)
- Differentiate your content delivery from the competition.
Why, why, why?

- The preferred methods for accessing information is expanding—**don’t risk losing or disengaging clients**.
- Audiences vary in terms of regulatory and other restrictions to content—including a variety of access points to **enhance their ability to learn**.
- That whiteboard on the right represents complexity that used to dictate a Faustian bargain—either go crazy, burn out, and deliver less, inferior content or **put our house in order**.
The payoff

• The HTML5 platform integrates well with source control—we use Subversion, which is open-source.

• An added benefit of using source control allows for multiple authors to create and update content—significantly increasing speed of delivery to market and resource bandwidth.

• Write once, apply conditions, variables, etc. and move on.

• Using CSS enables your clients (and anyone with an aesthetic) the benefit of content that has the same look and feel.

• The vendor we chose, MadCap (US), includes translation, feedback, graphics, and lots of additional value-add components.
First things first... the output

<table>
<thead>
<tr>
<th>Output</th>
<th>Easy to Use</th>
<th>Easy to Access</th>
<th>Oh Snap!</th>
<th>Easy to Distribute</th>
<th>Easy to Maintain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile (Phone/Tablet)</td>
<td>✓</td>
<td>Depends</td>
<td>✓</td>
<td>Depends</td>
<td>✓</td>
</tr>
<tr>
<td>Browser (HTML5)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PDF</td>
<td>✓</td>
<td>✓</td>
<td>Depends</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Can run in any mainstream commercial browser.

- Native apps
- Web apps
- Hybrid apps
Mobile: Native apps

- Follow a device or platform standard, such as iPhone apps or Android apps.
- Easily access on-device resources, such as a GPS, camera, flashlight, etc.
- Accessed or purchased via public or in-house app stores.
- Take advantage of the platform’s features, but limited by the platform.

For example, updating a native app for iterative OS releases can be time- and resource-consuming, and sometimes risky.
Mobile: Web apps

• Can run in any mainstream commercial browser. Note: Technically, WebHelp is a web app.
• Cannot easily access on-device resources
• Can be loaded on and accessed from a server rather than an app store.
• Easier to maintain and update than native apps because everything happens on the server side. Midnight updates, anyone?
Mobile: Hybrid apps

• Can appear in an app store and can access device resources, like a native app.
• Can run in a browser, like a web app.
• Can be extended with custom JavaScript or HTML.
• Leverages HTML5 and CSS3.
Mobile: Easy choice, right?

That depends:

- Our doc is not the product; it **contains information** to help clients use our products.
- For the product we piloted—it is fast-paced in terms of changes and updates.
- Our corporate web site does not yet support responsive web design; if we don’t, then clients are left with only PDF and rapidly deprecating WebHelp.
What is responsive design?

- A single version of your content
- Delivered to your clients (customers, patients, students, teachers, process partners, etc.)
- In a way that is easily understood
- No matter what device they are using
What responsive design is not...

- Coding exactly to every device on the market.
- Lessening the content.
- Providing silos of content based on device size.\(^2\)
Why HTML5?

- Fully supports responsive content, making it easier to create for the array of desktop, tablet, and smartphone users.
- It’s a simpler design approach and easier to incorporate audio and video content.
- Cleaner code enables authors to do the work of creating and maintaining content rather than figuring out how to invent a font.
- It’s fast and secure, in part because it stores data in the user’s browser (up to 5MB), alleviating the need for cookie deletion, third-party plugins, and other clunky distractions.³
Our journey with the complicated content

- Post-trade, pre-settlement software service called Omgeo Central Trade Manager® (Omgeo CTM).
- Global services across extensive country codes, currency codes, standards and regulatory entities, and the other factors.
- One writer (before you now) creating and maintaining content for 100+ business analysts, product managers, developers, integration teams, etc. (collectively *process partners*).
- Current library contains over 55 documents, manuals, and reference guides both client-facing and available by request only.
But wait, there’s more

Clients (Audience)
1. Investment managers
2. Broker/dealers
3. Interested parties

Interfaces
1. UI 4.0
2. Trade blotter
3. XML Messages
4. Financial eXchange Interface (FIX)
5. Message Translation Interface (MTI)

Asset Classes
1. Debt
2. Equity
3. Futures
4. Options
5. Repurchase Agreements (Repos)
Where the rubber hits the road

- Using MadCap Flare, in a project containing all of the content for Omgeo CTM, the Common Data Reference (CRD) is a rock star.
- The three *main* components of the CRD document are these:
Publishing the Common Reference Data

Clients
1. IMs
2. Broker/dealers

Interfaces
3. XML Msg Spec

Asset Classes
4. Debt
5. Equity
6. Futures
7. Options
8. Repos

Targets
- crd_html5.fltar
- crd_pdf.fltar

TOCs
- crd_html5.fltoc
- crd_pdf.fltoc

Contents
- Front/back matter, TOC, preface (print)
- Topics (chapters and sections) and images
- Styles (CSS) and reusable content/snippets
- Conditions (granular)

Cue the organ grinder...
Voilà—HTML5 output to browser and mobile x 2
Same content, conditionalized—some PDFs

It’s all in there, embedded in other docs (due to business requirements).
**Reality check**

- **How easy is Flare to use?** Like anything, it takes getting used to. I didn’t set up our templates, but if I need to tweak things here and there, I can. Converting documents to Flare is our biggest pain point.

- **Is it expensive?** It depends on how you view cost… are you or your staff constantly grinding out similar types of information but putting it in different documents and having trouble maintaining it? Or, occasionally losing versions of content? What if a key contributor wins the lottery?

- **How’s the technical support?** Phenomenal. Not only do my questions get answered, but they actively queue enhancement requests and file defects.

- **Is it widely used?** More and more.
References

Questions?
Thoughts?
Thank you!