

# Custom Font Usage in MadCap Flare Outputs

NOTE: updated version with answers to all follow-up questions -  
new slides added and enhancements woven in . . .

Wendy Studinski

*Certified MadCap Flare Consultant*

*Copyright January 2017*





# Wendy Studinski

Certified MadCap Flare Consultant

<https://www.linkedin.com/in/wendystudinski>

Flare power user since Flare 3.0 version.

Studinski Consulting Services: Helping your team succeed!

- Flare configuration
- Flare authoring

Past experience helping documentation teams around the world (Australia, China, England, Germany, Ireland, India, USA).

# Custom Fonts: *Art form since 1440s*

A powerful tool:

- For your *brand identity!*
- A *product sold* for others to use!
- Lots of possibilities!

# Custom Fonts within Flare

In a past assignment, the corporate logo was created with a custom font.

The documentation team was asked to use that font in the product documentation as well.

Meeting that request turned out to be a ***complex puzzle in Flare.***

I want to share that information with others.

# How does Flare use Custom fonts?

## *PDF output from MadCap Flare:*

Requires that your custom font be installed on the machine where Flare generates the PDF.

Then Flare embeds the custom font information within the PDF file.

# How does Flare use Custom fonts?

*Online output from MadCap Flare:*

***CAUTION:*** Unless you make additional modifications, *the custom font is missing when your Help files are viewed from any machine where the font is NOT installed.*

# What If?

What if you **install** the custom font on your customer's machine?

- Now your custom font is available to anyone who uses the programs installed on that machine!

*(Copyright violation!)*

- Many operating systems and programs require different font file formats, so your installation program must know what it is doing!

*(Complex install script!)*

# What If?

What if you **do not install** the custom font on your customer's machine?

- Now your custom font is not available within the programs installed on that machine!

*(Copyright protected!)*

- Your installation program is simplified!

*(Simplified install script!)*

# First 3 Steps to Protecting Your Copyright:

## 1. Understand Your Customer's Environment:

Which machines and browsers might your customers use to view your Help files? Which font file formats are required?

## 2. Determine the location of the required *group* of font files:

Include a *group of font files* in your Flare output  
= your custom font in the required file formats.

## 3. Identify this *group of files* in your CSS:

Your customer's computer and browser will automatically select and use the appropriate font file format when displaying your Flare Help.

# Step1: Which *Group* of Font Files?

- Operating systems require different font file formats:
  - Windows operating system requirements
  - MAC operating system requirements
  - UNIX operating system requirements
  - Etc., etc.
- Browsers can require different font file formats (for example \*.EOT, \*.OTF, \*.TTF, etc.)

# A few Font File extensions:

*.xxx	Font Format Type	*.xxx	Font Format Type
ABF	Adobe Binary Screen Font	PFA	Printer Font ASCII
AFM	Adobe Font Metrics	PFB	Printer Font Binary — Adobe
BDF	Bitmap Distribution Format	PFM	Printer Font Metrics — Adobe
BMF	ByteMap Font Format	AFM	Adobe Font Metrics
EOT	Embedded OpenType	FOND	Font Description resource — Mac OS
FNT	Bitmapped Font	SFD	FontForge spline font database Font
GEM	Graphics Environment Manager	SNF	Server Normal Format
FON	Bitmapped Font — Microsoft Windows	TDF	TheDraw Font
MGF	MicroGrafx Font	TFM	TeX font metric
OTF	OpenType Font	TTC	TrueType Font
PCF	Portable Compiled Format	TTF	
PST	PostScript Font — Type 1, Type 2	WOFF	Web Open Font Format

# There are many font file formats!

This website described 198 last time I checked:

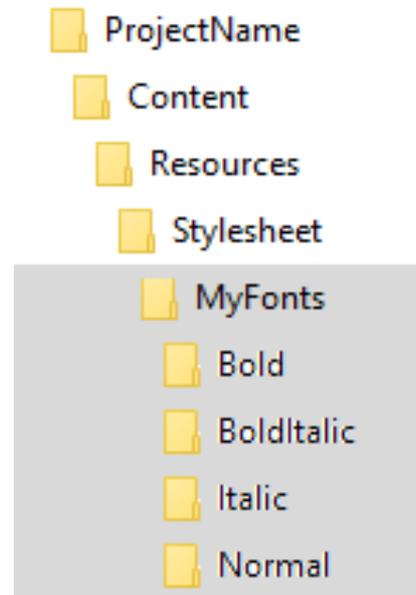
- <https://www.file-extensions.org/filetype/extension/name/font-files>

**CAUTION:** Educate yourself here.  
Give the font vendor your list of operating systems and browsers. Request font files to cover your entire list of needs.

## Step 2: Where to Place the Font Files?

Carefully place them within your Flare project's directory structure in a location that *ensures* they are automatically included within the Help output.

For example:



# Example Group of Font Files:

<ProjectName>/Content/Resources/Stylesheets/MyFonts/Bold/	MyFont-Bold.eot MyFont-Bold.ttf MyFont-Bold.woff
<ProjectName>/Content/Resources/Stylesheets/MyFonts/BoldItalic/	MyFont-BoldItalic.eot MyFont-BoldItalic.ttf MyFont-BoldItalic.woff
<ProjectName>/Content/Resources/Stylesheets/MyFonts/Italic/	MyFont-Italic.eot MyFont-Italic.ttf MyFont-Italic.woff
<ProjectName>/Content/Resources/Stylesheets/MyFonts/Regular/	MyFont-Regular.eot MyFont-Regular.ttf MyFont-Regular.woff

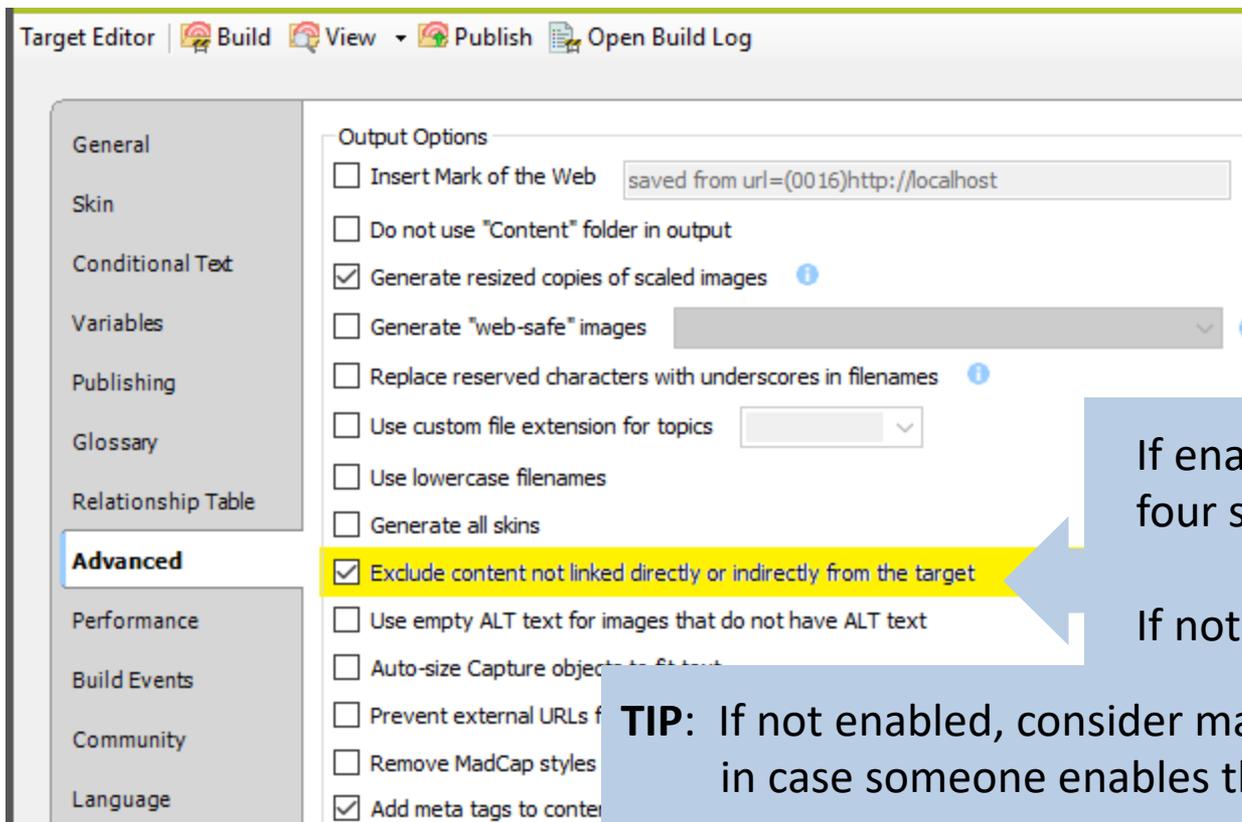
## CAUTION:

When your Flare output is *installed on a customer's machine*, the entire path + filename cannot exceed 260 characters.

If the 260 character limitation is exceeded, your font cannot be displayed.

# Check Your Target File

Look on the **Advanced** tab, and find the setting for *Exclude content not linked directly or indirectly from the target*



**NOTE:** Once *Enhancement Request 127420* is available, this will no longer be necessary.

If enabled, the steps on the next four slides are currently required.

If not enabled: they are optional.

**TIP:** If not enabled, consider making these changes anyway in case someone enables this Target file setting later.

# Background

## ***Exclude content not linked directly or indirectly from the target***

When enabled, Flare invokes a filtering script that excludes files from output unless they are:

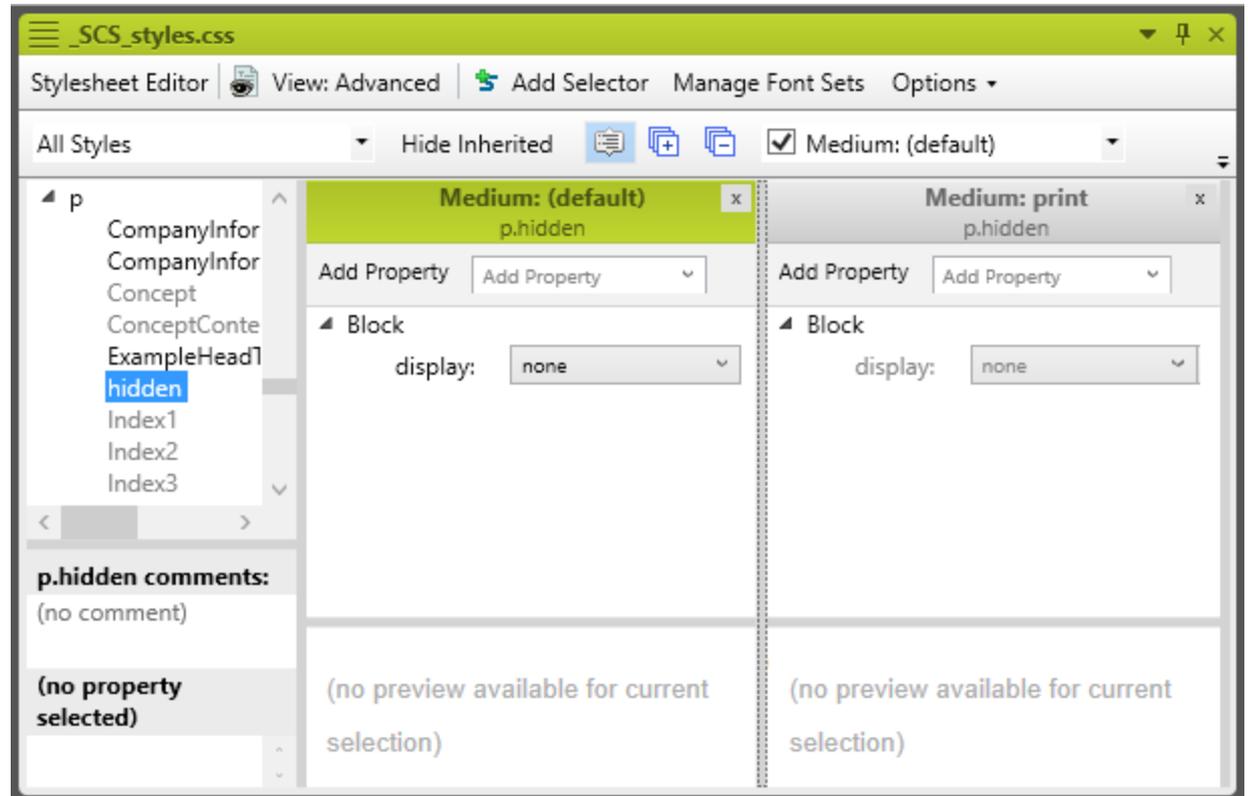
- Topic files listed in the TOC's urls
- Topic files connected to via a Hyperlink from above files
- Graphics that are used within one of above files
- Snippets used in the above files
- Only 1 skin file allowed (the one referenced in the Target file)  
**NOTE:** This can be a problem if you need one skin file for authorized users and a second skin file to prevent users without a valid *username & password* from accessing the program's Help TOC (such as clicking a login screen's Help button).

**CAUTION:** Test, test, test, because there is no comprehensive list available stating exactly what gets excluded.

# CSS Addition

Add a new <p> with the following settings:

```
p.hidden  
{  
  display: none;  
}
```



# Hyperlinks in a Topic File

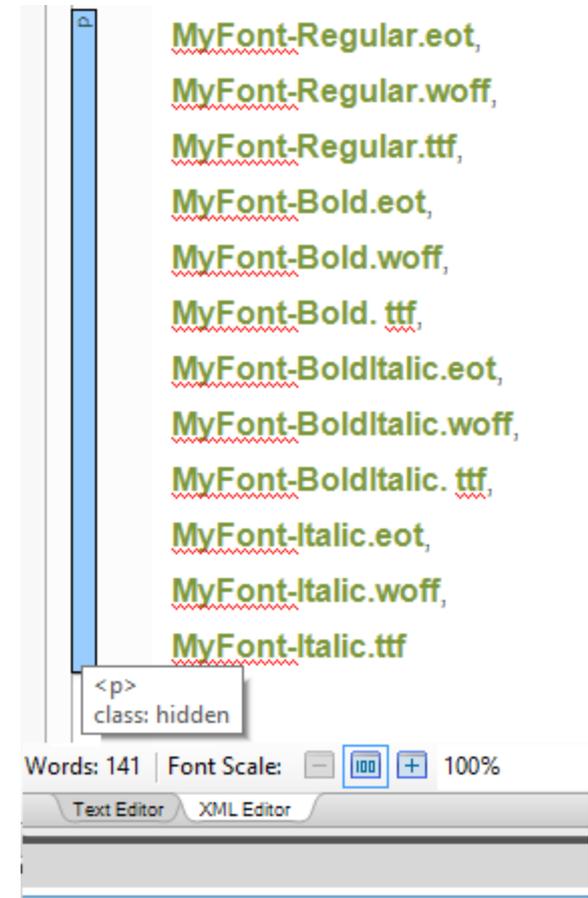
Choose a topic file.

Any topic file listed in the TOC will do.

At the bottom of the file:

- add one of your “hidden” paragraphs
- within that paragraph, add a **hyperlink** to each font file

These links are not visible in the Help output.



# Hyperlinks in a Topic File

These “hidden” hyperlinks ensure that Flare includes the font files in the Help output:

```
<p class="hidden">
```

```
<a href="Resources/Stylesheets/MyFonts/Regular/MyFont-Regular.eot">MyFont-Regular.eot</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Regular/MyFont-Regular.woff">MyFont-Regular.woff</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Regular/MyFont-Regular.ttf">MyFont-Regular.ttf</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Bold/MyFont-Bold.eot">MyFont-Bold.eot</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Bold/MyFont-Bold.woff">MyFont-Bold.woff</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Bold/MyFont-Bold.ttf">MyFont-Bold. ttf</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/BoldItalic/MyFont-BoldItalic.eot">MyFont-BoldItalic.eot</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/BoldItalic/MyFont-BoldItalic.woff">MyFont-BoldItalic.woff</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/BoldItalic/MyFont-BoldItalic.ttf">MyFont-BoldItalic. ttf</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Italic/MyFont-Italic.eot">MyFont-Italic.eot</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Italic/MyFont-Italic.woff">MyFont-Italic.woff</a>, <br />  
<a href="Resources/Stylesheets/MyFonts/Italic/MyFont-Italic.ttf">MyFont-Italic.ttf</a>
```

```
</p>
```

## Step 3: What CSS Changes Required?

To successfully display your custom font, you must understand and modify these settings:

1. @font-face
2. font-family
3. font-size

*% of em* versus *Point size* within:

@medium tablet

@medium mobile

@medium print

# *@font-face* Declarations = 4 parts

At the top of your CSS, provide declarations pointing to your custom font files:

```
@font-face
```

```
{
```

```
font-family: x;  
font-weight: x;  
font-style: x;  
src: x;
```

```
}
```



4 required aspects of *@font-face*

# ***CAUTION:* @font-face Declarations**

- The required additions at the top of your CSS ***cannot be created with Flare's internal Stylesheet Editor***. Right-click the CSS filename and it open in a text editor.
- You must use a text editor to carefully add the **@font-face** declarations to your CSS file.
- Use a text editor that does not change the line-ending characters when you save the CSS file!

## **For example:**

NotePad = good choice      WordPad = wrong line ending characters

# font-family

Choose a name for the font-family:

```
@font-face
{
    font-family: 'MyFont';
    font-weight: x;
    font-style: x;
    src: x;
}
```

Choose any name that is *different* from the font's "installed" name (the name used for PDF and print).

**NOTE:** This name does not need to match the directory name or font file names.

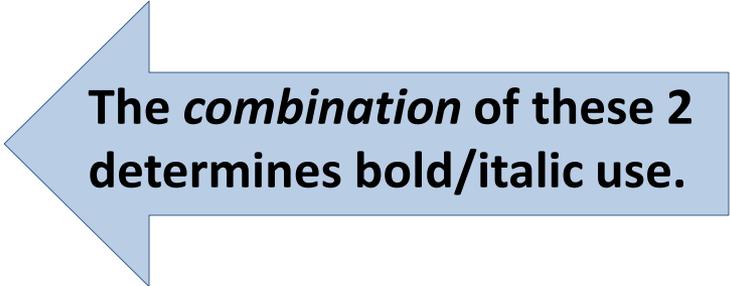
For example, the font-family name could be '1'

Anywhere the CSS *uses* those fonts, you refer to that font family:

```
font-family: 'MyFont';
```

# Weight and Style Combinations

```
@font-face
{
    font-family: 'MyFont';
    font-weight: x;
    font-style: x;
    src: x;
}
```



The *combination* of these 2 determines bold/italic use.

	Normal =	Bold =	Italic =	Bold Italic =
<b><i>font-weight:</i></b>	normal	bold	normal	bold
<b><i>font-style:</i></b>	normal	normal	italic	italic

# *src Statements*

**@font-face**

{

font-family: 'MyFont';

font-weight: normal;

font-style: normal;

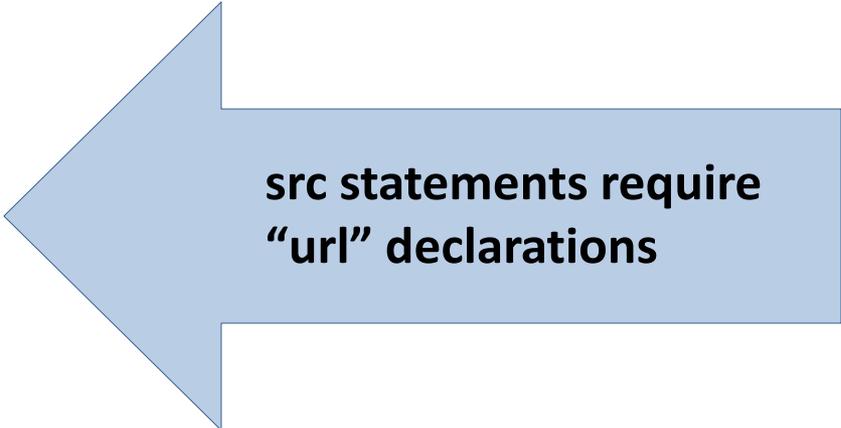
**src:**

**url (x) format (x),**

**url (x) format (x),**

**url (x) format (x);**

}



**src statements require  
“url” declarations**

# *src* Statements

*src's url* declarations tell Flare where to find your font files:

**src:** Path Statement (where to find each font file).      Type of font file.

```
url('./MyFonts/Regular/MyFont-Regular.eot') format('embedded-opentype'),  
url('./MyFonts/Regular/MyFont-Regular.woff') format('woff'),  
url('./MyFonts/Regular/MyFont-Regular.ttf') format('truetype');
```

Use *relative path statements* for movement from the CSS file's location to the font files.

# Relative Path Statements

Understanding *relative path statements*:

Where to look?	Meaning
./	<b>Move down</b> the directory tree from the CSS's current directory.
../	<b>Move up one</b> level in the directory tree from the CSS file.
../..../	<b>Move up the specified number</b> of levels in the directory tree

**CAUTION:** Always use the / *forward slash* (not backslash which is "Windows only").

@font-face

```
{  
  font-family: 'MyFont';  
  font-weight: normal;  
  font-style: normal;  
  src:  
    url('./MyFonts/Regular/MyFont-Regular.eot') format('embedded-opentype'),  
    url('./MyFonts/Regular/MyFont-Regular.woff') format('woff'),  
    url('./MyFonts/Regular/MyFont-Regular.ttf') format('truetype');  
}
```

1 Tab here

Flare uses this group of files for MyFont by default

@font-face

```
{  
  font-family: 'MyFont';  
  font-weight: bold;  
  font-style: normal;  
  src:  
    url('./MyFonts/Bold/MyFont-Bold.eot') format('embedded-opentype'),  
    url('./MyFonts/Bold/MyFont-Bold.woff') format('woff'),  
    url('./MyFonts/Bold/MyFont-Bold.ttf') format('truetype');  
}
```

Flare uses this group of files for “Bold” (for **words**)

@font-face

```
{  
  font-family: 'MyFont';  
  font-weight: normal;  
  font-style: italic;  
  src:  
    url('./MyFonts/Italic/MyFont-Italic.eot') format('embedded-opentype'),  
    url('./MyFonts/Italic/MyFont-Italic.woff') format('woff'),  
    url('./MyFonts/Italic/MyFont-Italic.ttf') format('truetype');  
}
```

Flare uses this group of files for “Italic” (for *words*)

@font-face

```
{  
  font-family: 'MyFont';  
  font-weight: bold;  
  font-style: italic;  
  src:  
  url('./MyFonts/BoldItalic/MyFont-BoldItalic.eot') format('embedded-opentype'),  
  url('./MyFonts/BoldItalic/MyFont-BoldItalic.woff') format('woff'),  
  url('./MyFonts/BoldItalic/MyFont-BoldItalic.ttf') format('truetype');  
}
```

Flare uses this group of files for “Bold Italic”

@font-face

```
{  
  font-family: 'MyFont_Light';  
  font-weight: normal;  
  font-style: normal;  
  src:  
  url('./MyFontsLight/Bold/MyFont-Light.eot') format('embedded-opentype'),  
  url('./MyFontsLight/Bold/MyFont-Light.woff') format('woff'),  
  url('./MyFontsLight/Bold/MyFont-Light.ttf') format('truetype');  
}
```

If more than one custom font, choose another name

@font-face

```
{  
  CAUTION: For the url statement, always use the /  
  forward slash (not backslash which is “Windows only”).
```

***CAUTION:*** See slides 15-19 for work around steps  
if your Target file has enabled:

***Exclude content not linked directly or indirectly from the target***

```
}
```

# Step 3: What CSS Changes Required?

To successfully display your custom font, you must understand and modify these settings:

1. @font-face

2. font-family

3. font-size

*% of em* versus *Point size* within:

@medium tablet

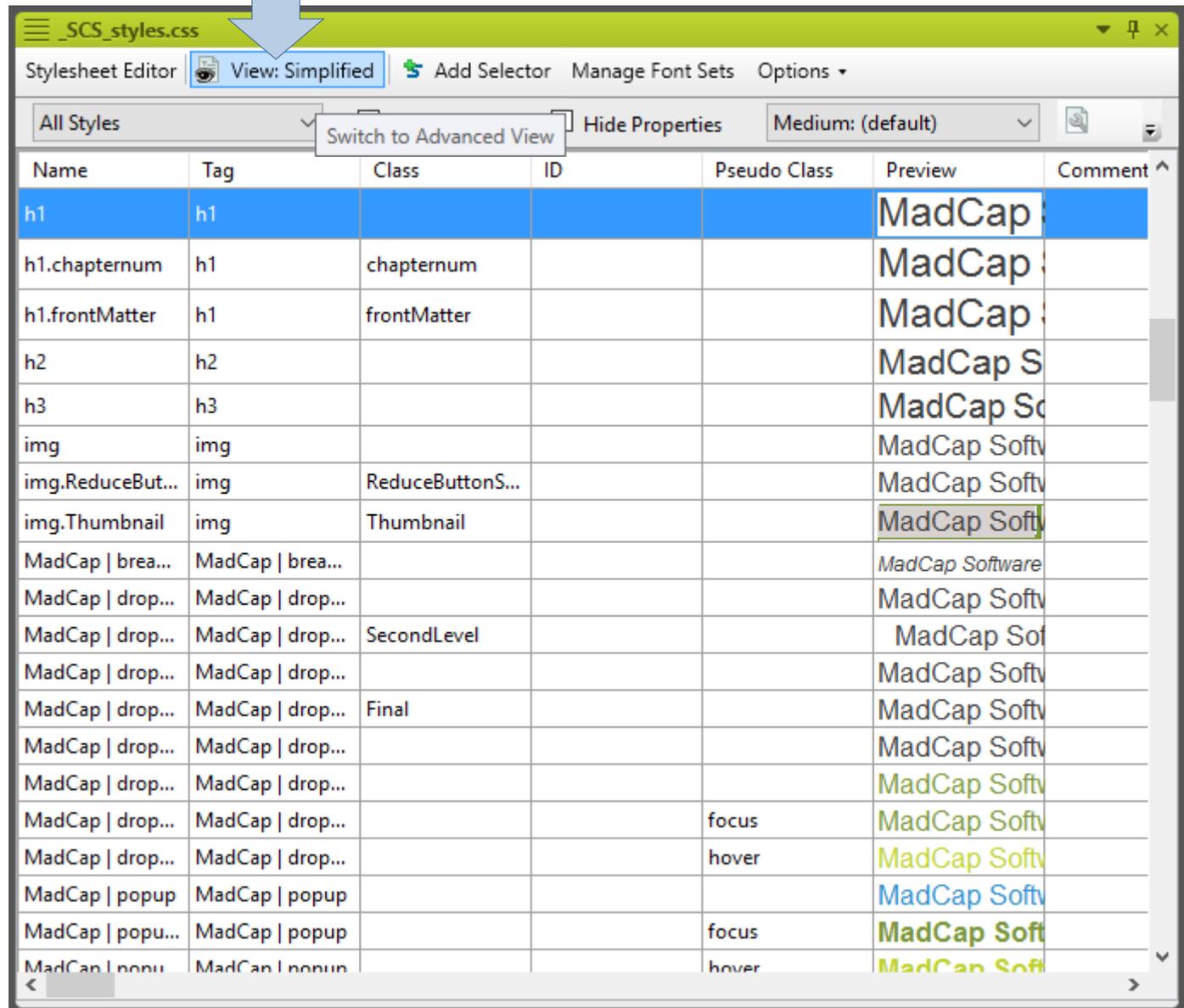
@medium mobile

@medium print

# CSS References to your Custom Font

- Yes, you *can* use Flare's Stylesheet Editor.
- Assign your custom font to various HTML elements.
- **@medium** = very important to understand these settings

# Flare's CSS Editor (use Flare 12 or +)



The screenshot displays the Flare CSS Editor interface for a stylesheet named "\_SCS\_styles.css". The interface includes a menu bar with options like "View: Simplified", "Add Selector", "Manage Font Sets", and "Options". Below the menu bar, there are controls for "All Styles", "Switch to Advanced View", "Hide Properties", and a font size dropdown set to "Medium: (default)". The main area contains a table of styles with the following columns: Name, Tag, Class, ID, Pseudo Class, Preview, and Comment. A blue arrow points to the "View: Simplified" button in the menu bar.

Name	Tag	Class	ID	Pseudo Class	Preview	Comment
h1	h1				MadCap	
h1.chapternum	h1	chapternum			MadCap	
h1.frontMatter	h1	frontMatter			MadCap	
h2	h2				MadCap S	
h3	h3				MadCap Sc	
img	img				MadCap Softv	
img.ReduceBut...	img	ReduceButtonS...			MadCap Softv	
img.Thumbnail	img	Thumbnail			MadCap Softv	
MadCap   brea...	MadCap   brea...				MadCap Software	
MadCap   drop...	MadCap   drop...				MadCap Softv	
MadCap   drop...	MadCap   drop...	SecondLevel			MadCap Sof	
MadCap   drop...	MadCap   drop...				MadCap Softv	
MadCap   drop...	MadCap   drop...	Final			MadCap Softv	
MadCap   drop...	MadCap   drop...				MadCap Softv	
MadCap   drop...	MadCap   drop...			focus	MadCap Softv	
MadCap   drop...	MadCap   drop...			hover	MadCap Softv	
MadCap   popup	MadCap   popup				MadCap Softv	
MadCap   popu...	MadCap   popu...			focus	MadCap Soft	
MadCap   nonu...	MadCap   nonu...			hover	MadCap Soft	

# font-family

Multiple fonts in list = safety net

Browsers try to use each font in the order they appear.

In (default) settings, add your custom font family-name first in the list.

Verify that all other @media “inherit” your custom font setting.

The custom font **is now available for the Help output files**. However if the custom font is not installed on the authors' machine:

- The custom font does not *display* in the CSS Editor.
- The custom font does not *display* within topic files.

**TIP:** Enhancement Request 127422 would resolve this issue.

# font-size + @media (% of em vs pt)

Each electronic device has a “default” font size setting that the device owner controls:

- If someone has good eyesight, their *default* font size setting can be quite small.
- If someone has poor eyesight, their *default* font size setting can be quite large.

This “default” font size setting makes the device compliant with Federal Disability Law 508 (making the device 508 compliant).

# font-size + @media (% of em vs pt)

There are four industry standard “Units” currently available to set font size:

**For example:** Device owner chooses *default 16px = 1em = 100% = 12pt*

- **Pixels (px):**  
1 px = one dot on the computer screen – the smallest division of your screen’s resolution. *The higher the resolution, the smaller the font.*  
The pixel unit is not dynamically scalable (*not 508 compliant*).  
Do not use this choice in your CSS settings.
- **“Ems” (em):**  
1 em = the device’s current default font-size.  
Ems are mobile-device-friendly and 508 compliant for accessibility.
- **Percent (%):**  
100% = the device’s current default font-size.  
% settings are mobile-device-friendly and 508 compliant for accessibility  
**DO NOT set to less than 100% (smaller than your customer’s choice)**
- **Points (pt):**  
1 pt = 1/72 of an inch (from printing press days).  
Points are used in PDF and when your Help topic is *printed on paper*.

Best choice for any online @media

Best choice for @media print

# font-size + @media (% of em vs pt)

Kyle Schaeffer wrote a fabulous explanation of font-size choices:

<http://kyleschaeffer.com/development/css-font-size-em-vs-pt-vs/>

When the browser or application uses font settings for Small / Medium / Large . . .

“em” causes problems, “%” works well:

body = 1em	Text-Size: "Smallest"	Text-Size: "Largest"
font-size: 1em	The quick brown fox jumps over the lazy dog.	The quick
font-size: 100%	The quick brown fox jumps over the lazy dog.	The quick br

© KyleSchaeffer.com

# font-size + @media (% of em vs pt)

@medium

(default)

tablet

mobile (phones)

print (PDF)

Verify that all @media groups have appropriate settings

font-size: In “online output” settings of your CSS, use font-size **100% (of em) or greater**

(100% = your customer’s default font size for 508 requirements)

Use the font name you chose for online @media

font-size: In the **@media print** settings of your CSS, use = **pt (points)**

(for PDF or to print Help topics)

Use the font name of the *installed* font.

# Font Size + @medium Statements

If you open your CSS in a text editor, partway down the page you will see:

(default) = any settings prior to the first @medium statement.

@medium tablet

@medium mobile

@medium print

Each group of @medium settings shows “***over-ride settings***” from (default).

# Font settings in your CSS:

```
h1
{
  font-family: MyFont, Arial, Tahoma, 'Lucida Sans Unicode', sans-serif;
  color: #003366;
  font-weight: bold;
  font-style: normal;
  font-size: 150%;
  line-height: normal;
}
```

font-size

```
@media tablet
{
  h1
  {color: #0000cd;}
}
```

```
@media mobile
{
  h1
  {color: #006400;}
}
```

```
@media print
{
  body
  {font-size: 11pt;}
```

font-size

font-size

```
h1
{
  color: #7b973c;
  font-family: MyFont , Arial
  font-size: 26pt;
  margin-top: 28pt;
  margin-bottom: 14pt;
  page-break-after: avoid;
}
```

**font-size:** In the “online output” sections of your CSS, use font-size = **100%** (of em) **or greater** (100% = your customer’s default font size for 508 requirements)

**font-size:** In the **@media print** section of your CSS use font-size = **pt** (points for PDF or to print Help topics)

# Using your custom font:

/\* To use these fonts, use

**font-family:** Your customer's machine moves down the list to find an available font.

h1  
{

font-family: **MyFont**, Arial, Tahoma, 'Lucida Sans Unicode', sans-serif;  
color: #003366;

Which group? →

**font-weight:** bold;  
**font-style:** normal;  
font-size: 150%;  
line-height: normal;

← **font-size:** carefully choose a % of em (for online) or point size (for print)

}

h2  
{

font-family: **MyFont**, Arial, Tahoma, 'Lucida Sans Unicode', sans-serif;  
color: #003366;  
font-weight: bold;  
font-style: normal;  
font-size: 110%;  
line-height: normal;

}

... Or font-family: **MyFont\_Light** in this example

# Two Additional Concerns

Step 4: Skin File Addition

Step 5: Encryption

# Step 4: Skin File Addition

## For WebHelp or HTML5 Help

Your CSS file changes allow the custom font to display within the *topic frame*.

To display your custom font in the skin file's *banner frame* and *TOC frame*, additional changes are required.

# See Dave Lee's description of css and JavaScript modifications to the skin.

Search the MadCap blog for Dave Lee's information about applying your updated CSS to the skin file:

Here is Dave Lee's project about how to customize an HTML5 skin:

- <http://ukauthor.esy.es/SkinMods/Default.htm>

Dave Lee's instructions for adding your *stylesheet* to the skin is here:

- <http://ukauthor.esy.es/SkinMods/Default.htm#Skin-Stylesheet.htm>

# Dave Lee's <http://ukauthor.esy.es>

ukauthor.esy.es

## Dave Lee - ukauthor.esy.es

Flare e

- Exam
- Exam
- Exam

The screenshot shows a web browser displaying a page from ModCap. The page title is "Customised skin". The breadcrumb trail is "> Customising the HTML5 skin > Examples > Add a stylesheet to the skin". The main heading is "Example 2: Add a stylesheet to the skin". The text explains that the stylesheet can be used to style content inserted in the skin, such as the header, and to modify the appearance of elements that can't be changed using the skin, like the width of the responsive menu. It notes that simply adding a stylesheet link to the skin output file (Default.htm) is not the correct approach. Therefore, the example shows how to use a toolbar script to dynamically insert a stylesheet link. The steps are:

1. Where do I enter the skin toolbar script? ▾
2. To add a stylesheet link to the skin output file, enter this code:

```
$('<link>')
    .appendTo($('head'))
    .attr({type: 'text/css', rel: 'stylesheet'})
    .attr('href', 'Content/Resources/Stylesheets/skin.css');
```

# Step 5: Protect the Font Files

Ask your R&D team to take best practice measures to minimize chances of customers:

- Finding the font files.
- Copying and installing the font files anywhere without permission.

**TIP:** Share the next slide with your R&D team.

# Best Practice: Protect the Font Files

**Help output provided by an application in the computer's local file system:**

A local application could encrypt the help files and/or custom font files, but then the application's R&D team (for whom the Help is written) would need to create decryption scripts to display the fonts.

**NOTE:** When the Help files are displayed via a browser, an unencrypted copy is made in the browser's temp setup. Any way to automatically delete those?

**Help output installed on a server and accessed via web applications:**

Font files cannot simply stay encrypted; otherwise, the browser wouldn't be able to read them. Preventing a sophisticated user from copying/using your custom font can be accomplished by either:

- using a font hosting company, which provides this as part of its service, or
- asking your R&D team to design the application to obscure/obfuscate the Help's font files, making them *harder* to steal them

**NOTE:** This is different from SSL encryption, which encrypts while transporting over the wire, then unencrypts the copy at the destination.

For more information, see:

- <http://typedrawers.com/discussion/91/preventing-download-of-web-fonts>
- <http://stackoverflow.com/questions/8907036/how-can-i-prevent-web-fonts-from-being-downloaded-and-used-illegally>
- <https://blog.typekit.com/2009/07/21/serving-and-protecting-fonts-on-the-web>

# Review

1. Understand Your Customer's Environment (Which group of font file formats?)
2. Determine the location of the required group of font files in your Flare project and check your Target file settings.  
(Advanced tab's: Exclude content not linked directly or indirectly from target)
3. Identify this *group of files* in your CSS
  - a) @font-face declarations (4 parts within each) – use a text editor
  - b) font-family (installed font name for print, your chosen name for online)
  - c) font-size: % of em versus *Point* size within:

<ul style="list-style-type: none"><li>• (default)</li><li>• @medium tablet</li><li>• @medium mobile</li></ul>	<b>100%</b> or greater = 3 online formats
<ul style="list-style-type: none"><li>• @medium print</li></ul>	<b>pt</b> = Printed Help topic and PDF format

4. Skin File Addition
5. Best Efforts to Protect the Font Files

# Success!

- Your customers see the custom fonts in your PDF file.
- Your customers see the custom fonts in your Help files, no matter where your Help files are installed (web servers or local machine).
- You have made “best efforts” to ensure your custom fonts won’t be hacked or used without permission.

# Conclusion

Custom fonts are a valuable tool within Flare if you know how to use them wisely.  
Best of luck in your authoring adventures!!

Wendy Studinski

*Certified MadCap Flare Consultant*

## TAKE ADVANTAGE OF CERTIFIED TRAINING

As a webinar attendee, receive **\$100 OFF** your next advanced training course.

---

### **MadCap Flare CSS Training**

January 23-24, 2017 (web-based)

### **MadCap Flare Single Sourcing Training**

January 26-27, 2017 (web-based)

For more details, [click here](#) or email [sales@madcapsoftware.com](mailto:sales@madcapsoftware.com)

*Note: Courses subject to change. Availability based on student registration. Certain restrictions apply; cannot be combined with any other offer or promotion. Not valid on courses already purchased.*

# MADWORLD

*The Premier Technical Communication  
and Content Strategy Conference*

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

*Register by January 31 and Save Up to \$500*

**[WWW.MADCAPSOFTWARE.COM/EVENTS/MADWORLD](http://WWW.MADCAPSOFTWARE.COM/EVENTS/MADWORLD)**

# Contact Information

- <https://www.linkedin.com/in/wendystudinski>
- [studinskiw@gmail.com](mailto:studinskiw@gmail.com)

Wendy Studinski

*Certified MadCap Flare Consultant*