

Updating the Sony Icon Font

A Front End Developer's guide to updating the Sony Icon Font

Icon Font Concept

Icon fonts are no different from regular fonts, except that they use icons instead of letter shapes. They are implemented using the *@font-face* css property, and rendered on the page using css pseudo-elements.

Advantages

There are several advantages of using an icon-font over spritesheets:

- Efficiency: Single http request for all icons. It can even be combined with other fonts in a single css file with base64 encoded fonts for fewer http requests (which is what we're doing)
- Versatile: Can be styled with css, including color, size, and even using css transitions.
- Scalable: No need for a separate 'retina' version.

How They Work

As an example, we could make the lowercase letter "a" look like a heart icon. By setting the css font-family to use our icon font, anywhere the letter "a" appeared would render as a heart. If we mapped all our icons to actual letters, it would lead to lots of random letters scattered across the site, and that's not good for SEO or screen readers. To better hide the fact that we're using letters to symbolize icons, we don't map our icons to regular ("Basic Latin") letters. Instead, we map our icons to the "Private Use Area" of the font. This is an area of a font that most fonts don't utilize, but that is accessible by all modern browsers using the Unicode string for each character.

Yes, it's a bit convoluted.

Icon Fonts are great, but the process of creating and updating them isn't exactly streamlined yet. Until better tools are available, this is what we've got. If you run into problems, just go through the process again step by step to make sure you didn't miss anything.

So instead of mapping the favorites icon to the letter "a" we can map it to the unicode character "\e013" for example. Just like normal fonts progress through the alphabet (a, b, c, etc.), our icon font will progress through these unicode characters (\e000, \e001, \e002, etc.).

We implement the icons using css classes, and insert the unicode characters using pseudo-elements, so they're not actually in the HTML markup. By implementing the icons based only on a class name, even if the font changes, and an icon is assigned to a different unicode character in the font, it will still render the correct icon as long as we also update our css to coincide with the new character mapping.

IcoMoon and Source Files

We'll be using the IcoMoon.io web app to build the actual font file. The site also offers an array of off-the-shelf icon fonts, but we're just going to use their font-building app.

The app takes a folder of *svg* files and exports several formats of web fonts. The site doesn't have accounts, but it does allow you to export a *json* file with your current session, including all the icons, and load it later. It's important to save an updated version of this *json* file any time the font is updated, so the font doesn't have to be recreated from "scratch" next time.

The entire font can be recreated using nothing but the Illustrator files, so it's very important to always keep them up to date with any icon additions or changes. They should be considered the "master" source files for the font. The *json* file is the secondary source, and we'll usually use that so we don't have to re-import all the icons. The *svg* folder can also be used as the source. All three should always be up to date in the most recent version of the font folder, and checked into the repository.

Icon Sizes

Icon fonts are vectors by nature, but unfortunately when scaled at small sizes their edges get blurry and don't render consistently. To

compensate for this, it's best to create icons at the exact size they'll be used at, and to align them to the pixel grid when designing them. If you need two sizes of the same icon, you're best off making two different versions of it – one at each size.

For our font, we're using multiples of 8 for icon canvas sizes. For example, an icon with dimensions of 12px by 15px should be put in a 16px canvas (the canvas is always a square), and displayed at 16px (1.0em) font size. An icon with dimensions of 10px by 18px should go in a 24px square canvas and be displayed at 24px (1.5em) font size, and so on. The canvas sizes we've defined for the Sony font are:

- 16px (1.0em)
- 24px (1.5em)
- 32px (2.0em)
- 48px (3.0em)
- 64px (4.0em)
- 80px (5.0em)
- 96px (6.0em)

There is a separate Illustrator file for each icon size. If adding an icon, be sure to add it to the appropriate Illustrator file.

Updating the Font

Creating & Preparing Icons

For details on designing icons for the site, see the Style Guide Iconography page <http://sandbox.odopod.com/sony-style-guide/doku.php?id=design:guidelines:iconography>

Before exporting, make sure the icons:

- ARE in the appropriate icon file for their size.
- ARE in the "icons" Illustrator layer, and named with the root class name you'd like the icon to use; for example, "favorites". (Also see

the note in the illustrator file about offset paths for bolder/lighter fonts).

- ARE aligned to the pixel grid (as well as can be - icons with no straight edges are mostly arbitrary). See Style Guide for details.
- Are NOT Compound Shapes -- Use Expand, Unite, etc. to make them into Paths or Compound Paths (not Compound Shapes!).
- Are NOT using any Appearance effects (such as stroke, rounded corners, warp, etc.). Use "Expand Appearance" to apply effects to the shape.
- Are NOT using multiple colors or transparency. Fonts can only be one color. The fill color in the Illustrator doesn't actually matter.

Preparing the Directory

Before you get started, you should create a new Fonts folder to hold the new assets:

- There should be an existing *Sony Icon Font #.#* directory with the most recent version of the font source files in it, in the repo:
sonyglobal/development/resources/icon-font/2-IcoMoon/
- Duplicate the most recent folder and increment its version number. This is our new Font folder.
- In the new folder, delete everything except the *svg* folder. If you're replacing an icon, delete the old one from the *svg* folder.

Exporting Icons

Install the *Make-Icon-Font-Assets* Illustrator script:

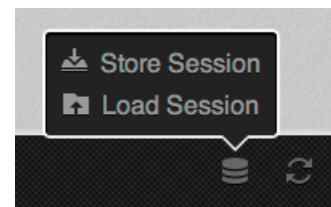
- from the repo: *sonyglobal/development/resources/icon-font/1-Illustrator-Source/!Make-Icon-Font-Assets--Illustrator-Script.jsx*. On Mac, it goes in */Applications/Adobe Illustrator CSX.X/Presets/en_US/Scripts*.
- This script runs through a series of automated actions to create individual *svg* files from the icons in the Illustrator file. For each icon, it creates a new Illustrator file, places & resizes the icon in

the proper position on the stage, applies a small *inset path* to "lighten" the icon for better cross-browser rendering, and then exports it as an *svg*. To use the script:

- Open the Illustrator file you're exporting, select the icon(s) you want to export, and run the *!Make-Icon-Font-Assets* script from *File > Scripts >*.
- Select the *svg* folder inside of your new *Sony Icon Font #.#* folder as the destination.
- If you have multiple sizes of icons, do this for each Illustrator file, into the same *svg* folder.
- When the script is done there will be one Illustrator temp file (*XXXXXX._t_*) in the directory for each exported icon. You should delete them.
- Note: The script has been tested on Illustrator CS5.1 and CS6.

Building the Font with IcoMoon.io

1. Go to <http://icomoon.io/app/>
 - Note: If your computer/browser were used to create the most recent version of the font, IcoMoon should remember your session. It's probably best to re-import the settings anyway, in case someone else updated the font in the mean time.
2. Select **Load Session** from the cylinder icon menu at the bottom-right of the page, and select the *Sony-Icons-Settings--IcoMoon.json* file from the most recent Fonts folder.
 - From the repo: *sonyglobal/development/resources/icon-font/2-IcoMoon/Sony Icon Font #.#*. This will import the font settings and all the icons from the current version of the font.
3. Click the **[Import Icons]** button. Select the icon(s) you want to add.
4. Use the **[select]** tool to select the new icons.
5. Click the **[Font]** button at the bottom of the page. Your new icons



should appear at the end of the list, with a higher unicode number than the rest.

6. Because we imported the json settings, the Preferences and Font Metrics should already be set. For reference, they should be:

☒ Set Font Metrics Manually

Em Square Height

Baseline Height % Em

Whitespace Width % Em

7. Select **Store Session** from the cylinder icon menu. Save the file in your current Font folder.
8. Click **Download** to download a zip with your compiled font files. Save it to your current Font folder & unzip it.

Applying the Update on the Site

1. Update the EOT font file:
 - Copy the `/fonts/Sony-Icons.eot` file to `/src/packages/common/fonts/`. That's all you need from the `/fonts/` folder.
2. Update the IE7 Polyfill:
 - Open the `lte-ie7.js` file, and copy its entire contents into `/src/packages/common/js/libs/polyfill-lte-ie7/lte-ie7-fonticons.js`, replacing all existing content.
3. Update the WOFF & TTF css font sources:
 - Open the `style.css` file, copy the base64 string for the `woff` font, and paste it into the same place in `/src/packages/common/fonts/b64-fonts-woff.css` (it should be the last font source in the file), and also into `/src/packages/common/fonts/b64-icons-woff.css`.
 - Do the same for the `ttf` font string, into `/src/fonts/b64-fonts-ttf.css` and `/src/fonts/b64-icons-ttf.css`.
4. Update the css classes & unicode mappings for all the fonts:

- Open `src/packages/common/css/_base/_fonticons.scss`.
- There's a big list of fonticon classes below the **IcoMoon Classes** **Go here** comment. Look at their structure. You're going to be replacing all of these with an updated version from the exported font file, but you'll need to add the `--after:after` lines yourself.
- From `styles.css`, copy all of the classes, starting with the first instance of `.fonticon-10-iconname:before` through the end of the file, and paste them over the classes in `src/packages/common/css/_base/_fonticons.scss`. You might want to keep the first one as an example, and delete it when you're done.
- Note: We're updating *all* of the classes, rather than just the one(s) just added, so that in the event the new icons changed the unicode character mapping of the old icons, the classes will still be mapped to the correct unicode character in the new font.
- For each class, add a line for an `--after:after` version; for instance, for

```
.fonticon-10-iconname:before,
```

add the class

```
.fonticon-10-iconname--after:after
```

- In SublimeText2, you can do this quickly by:
 - Select the first instance of just this text: `".fonticon-"`
 - Use *Quick Add Next* (**cmd+d**) to add each next instance of `".fonticon-"` to the selection. Go all the way down to the bottom of the page, being careful not to loop back around to the top where other classes have the same prefix.
 - Select the entire line (lines), copy/paste all at once as you need to make the edit to all the classes at once using multiple cursors.
 - You should now have this, for each class:

```
.fonticon-10-iconname:before,
```

```
.fonticon-10-iconname--after:after {
```

5. Update the Style Guide Icons:

- Open *index.html* (from the exported font folder).
- At about half way down the page source, after `<h1>Class Names</h1>`, copy all of the `` sections.
- Replace all instances of

```
<span class="box1"> with <li>
<span aria-hidden with <i aria-hidden
></span> with ></i>
</span> with </li>
```

- Now they should look like this:

```
<li>
    <i aria-hidden="true" class="fonticon-10-iconname">
    </i>&nbsp;fonticon-10-iconname
</li>
```

- Copy and paste these into the appropriate places in the Style Guide, replacing the existing list. It's in two places:
- <http://sandbox.odopod.com/sony-style-guide/doku.php?id=development:globalstyling>
- <http://sandbox.odopod.com/sony-style-guide/doku.php?id=design:guidelines:iconography>

Updating the CDN

The fonts are hosted on our own fonts.sony.net CDN, so we need to update them there as well. The copy in the site is really just to keep things organized (and to fall back on when testing).

1. Copy the following files to the CDN:
 - b64-fonts-ttf.css
 - b64-fonts-woff.css
 - b64-icons-ttf.css
 - b64-icons-woff.css
 - Sony-Icons.eot
2. We've been versioning them because they're cached and we want to make sure our css and our fonts are online at the same time. So

on the CDN, create a new incremental folder (ex: /v7/) at *http://fonts.sony.net/shf/* and copy the font files listed above to it.

3. Since we updated the fonts folder, we have to update our paths to the fonts. Update the following files with the new path (in 2 places in each file):
 - /sonyglobal/development/src/packages/common/html/fonts.html
 - /sonyglobal/development/src/packages/docs/html/docs-head.html
 - /sonyglobal/development/src/packages/modules/universal-nav/html/universal-nav-standalone-fonts.html
 - /sony-style-guide/lib/tpl/sony/main.php
 - /sony-style-guide/lib/tpl/sony/mediamanager.php