

# Creating Custom Font Packs

## Glyphter

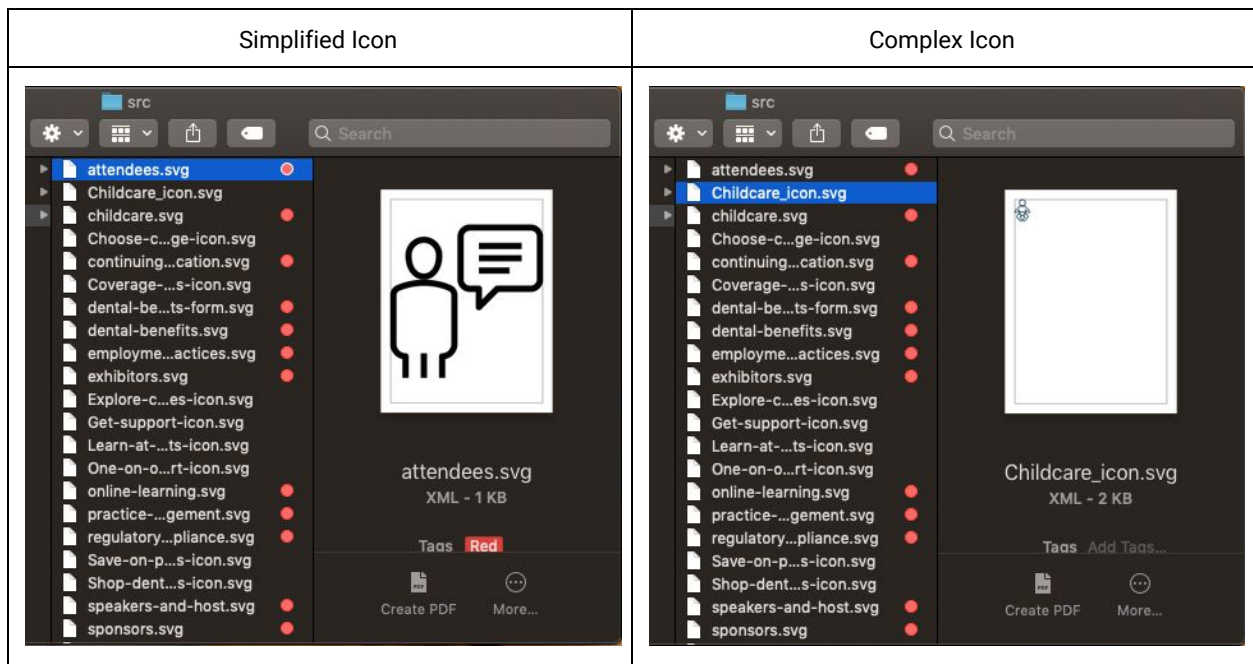
We use Glyphter to create our font packs. It provides a fairly straightforward way of creating icon packs for scalable website icons. Website: <https://glyphter.com/>

You won't need an account to create a font pack, however they do limit the amount of icons you can create in a pack for free users. If you reach the limit it may be worthwhile to look at getting a subscription to create the pack.

## Ensuring compatible svg formats

When creating font icons, glyphter can have issues creating icons if they are too complex. The best way to ensure the complexity of the icon is minimized is to fully expand the elements in the icon and merge all elements into a single line object if possible.

You can easily tell if glyphter will have issues creating the font icon by looking at it in your finder. When an icon is considered complex, it will not take up the full width of the preview. See the below screenshots.



## Reformatting icons

The easiest way to ensure icons are compatible for glyphter is to:

- Open up the svg file in Illustrator
- Select the entire object
- From the top toolbar select **Object > Expand...**
- From the Expand dialog, leave all the options alone and select **OK**

- Then open the Pathfinder window (From the top toolbar select **Window > Pathfinder**)
- Select the first Shape Model: **Unite**
- Once done, save the svg file, and you should see the icon now take up the full width of the preview. This icon is now suitable for glyphter.

### Importing to Glyphter

Importing the icons to Glyphter is fairly easy, but the most important aspect is ensuring that your icon order is consistent. Glyphter uses a consistent naming convention for the css class assigned to each icon. (Icon #1 should always be {content:'\0041';}). To ensure that the icons don't change when updated on the website, you will need to ensure that the icons are added in the same order they were originally.

Our normal practice is to always upload the icons alphabetically. In the event you need to update your icon pack multiple times, you will want to re-order your icons to match the updated version, while still remaining alphabetically per the update group.

We've created a few release directories in /src/ which will help you organize fonts by release version and make sure they are imported in the correct order into Glyphter.

You can then select each release and drag and drop each .svg into Glyphter. You can drag an entire group and drop it into the first available spot, and it will auto assign them based on name and order.

Before exporting the font icon pack, you will need to update the naming convention. From the top left, select the gear icon. From the popup, update the Class Prefix field to be "glyphter-".

### Exporting from Glyphter

Once you have all of your icons in the system and are sure that they are displaying correctly, you can export the pack.

- Click the font button at the top of the page with the download icon
- Click Just Download from the popup dialog
- Save the archive somewhere on your machine and extract it once downloaded
- Once opened you should see 2 directories; CSS and Fonts.

The next best practice would be to run a comparison between the new CSS file and the existing css font file in your repository or on the server. Compare the 2 files and ensure that all previous font icons are matching up correctly for the content:'x'; values.

Once verified you can add the new CSS and Fonts directory to the font directory located in:  
/site/wwwroot/Portals/\_default/Skins/[Your Skin Pack]/css/glyphter

[I would suggest creating a local backup of your css and font directories to ensure you have a local backup if you need to revert a change quickly.](#)

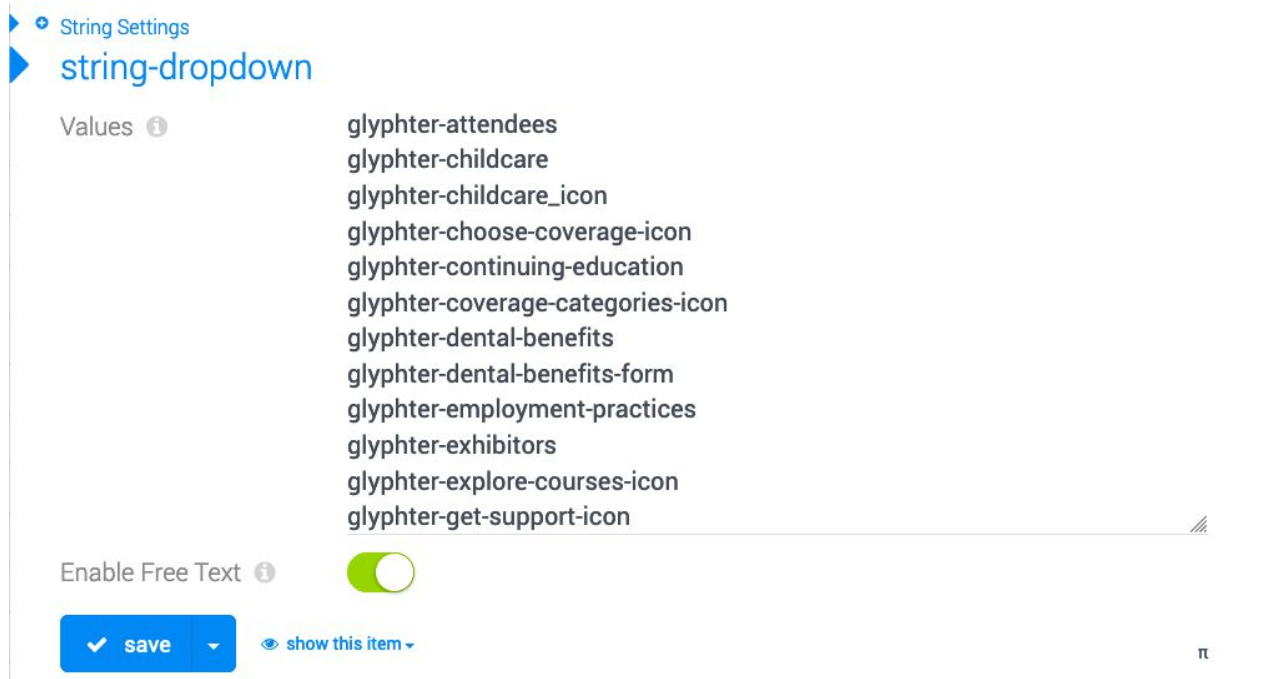
Update both your repository and your files on the server and you should at this point, not notice any changes. If you reload a page with existing fonts, you should notice that all existing fonts match up. If not, you will need to go back and review which font icons are off (in your comparison) and re-order and re-build if necessary.

## Updating 2sxc Icon Entity

- When in Edit mode, go to any 2sxc component and from the DNN edit pencil, select **Admin**.
- From the 2sxc admin popup, Click to open the **Data** tab.
- Search for the **Icon** Content Type and click on the Fields button.



- Click the text IconClass to open up the dialog to edit the icon drop-down list.
- When editing the field, you should see a list of fields, which are the class labels displayed when selecting an icon to use.



- Add your new icons, preferably in alphabetical order. Be sure to add the prefix glyphter- as used in the css. Also ensure that capitalizations for values are removed.
- Once done, save the edits and back out of the edit controls.

Now you can go to any element that is using the Icon entity, and adjust the icon to use any of the new icons. If an icon does not show up once selected, that is an indicator that the value in the 2sxc dropdown does not exactly match the class in the glyphter css file.