

AMERICAN CAPTAIN

ABCDEFGHIJKLMNOPQRSTUVWXYZ



Designing a font

Mike Adkins



About the author

Mike Adkins might just live as far away from the major design centers of the world as possible, in the small town of West Siloam Springs, Oklahoma. He graduated from the graphic arts program at Northeastern State University in Tahlequah, Oklahoma, but the time was the early 90s, when computers were still a rare commodity and graphic design jobs were hard to find locally. Unable — and stubbornly unwilling — to relocate, he did the next best thing: he bought a computer and a vinyl cutter and started a home-based sign shop, tackling everything from banners to vehicle graphics. As a natural extension of years of working in CorelDRAW, he continued using it as his main artistic tool when he began to dabble in font design. By the mid-90s, the Fontry was born, offering typographic services worldwide through the Internet. Visit the Fontry's website at www.thefontry.com.

Designing a font

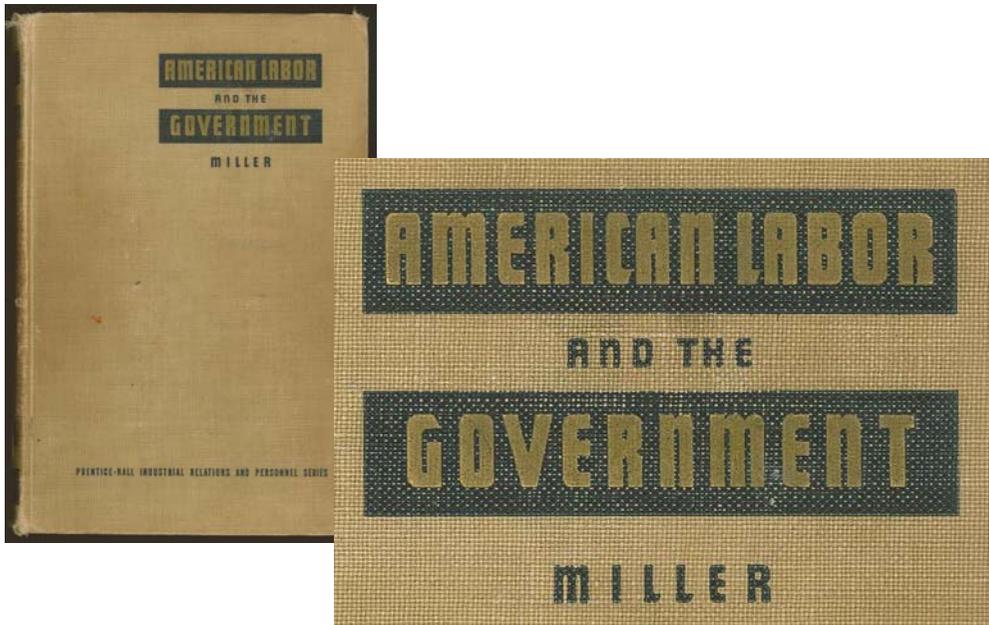


Figure 1: Vintage lettering

Have you ever found yourself staring a little too long at a scribbled word on a piece of paper? Or lost in thought while studying the faded lettering on an old hand-painted sign? Or thumbing through some vintage literature one day when it hits you (Figure 1)? Those letters staring back at you from an old book cover have suddenly stirred your creative spirit, becoming mental images of shapes and forms that want to find their way out of your imagination and onto computer keyboards around the world!

As dramatic as all that sounds, I should begin by asking an obvious question. So you want to design fonts? You've been bitten by the font-

making bug and suddenly you feel it's your destiny to make real all those wonderful letterforms that are trying to claw their way out of your head. Now you have many questions. Or maybe just one question — the biggest question of all.

Where to begin?

I've been making fonts for almost two decades, and twenty years ago, I asked myself the same question. Then, as now, font-making advice was hard to come by, and when found, it was (and still is!) downright vague and mysterious. A single tutorial cannot turn even the most experienced and computer-savvy

designer into an expert font-maker. However, whether you're an expert at type design or a complete amateur, I hope that the lessons shared here will offer you a bit of worthwhile guidance through the most important stages of the process. They will also explain how a graphic design program like CorelDRAW X7 can play an indispensable role in your font-making endeavors.

Keep in mind, however, that CorelDRAW is only the artistic component in the whole process. In order to compile and generate a font, you also need a dedicated font editor that can handle the idiosyncrasies of typography. There are many good programs out there, even some free ones. I personally like Fontlab, but no matter which one you choose, they all have a steep learning curve. So be ready to spend some time getting familiar with the software.

In this article, I'll delve only into the basic aspects of getting a font ready for the font editor by using nothing but a scanner, a mouse, and CorelDRAW X7 as the primary design tools.

Getting started

There are two basic facts about font design.

Fact #1: A font is a collection of vectors (Figure 2) — outline shapes that are defined by dynamic control points — all arranged to represent an alphabet that can be keyed in and realized digitally on everything from a cell phone to a movie screen.

Q: What is a font really?

A: Vectors and Control Points!

Figure 2: A font is a collection of vectors.

If you truly want to make fonts, prepare to create vectors, and lots of them! While most

font editors are also stand-alone vector-based drawing programs, they have their limitations. A dedicated illustration program like CorelDRAW can greatly enhance your precision control and creativity. When the final vectors are ready, you can export them from CorelDRAW as files that can be seamlessly imported into your font editor.

Fact #2: The overall look and feel of the alphabet is a critical starting point, so every type-design effort should begin with a basic concept or theme. Will the font be narrow? Tall? Thick? Thin? Futuristic? Something with a vintage, antique feel? Serifs? No serifs? The choices are many. Once you have an idea in mind, a good way to start is to shape your font around a word or a unique set of letters.

For this lesson, I will share with you some of the techniques that went into the making of the American Captain font (included with CorelDRAW X7). In trying to decide how this font would look, I focused the entire concept around the word CAPTAIN — all upper-case — with careful attention to the final arrangement of the letters in AMERICAN (Figure 3). This entire font was basically built around the appearance of the nine letters in those two words. This approach of focusing only on a few letters in a word or two can make the effort of designing an entire alphabet far less overwhelming.

I also like to start all my fonts by designing the capital letters first. This allows the rest of the font to build itself up naturally around the finished upper-case range of characters, in particular the numbers and much of the punctuation, and ultimately the lower-case letters.

AMERICAN CAPTAIN

ABCDEFGHIJKLMN**OP**QRST**UV**WXYZ

Figure 3: The letters highlighted in red represent the nine letters that spell out the font's name.

For American Captain, I envisioned a narrow, yet bold, non-serif block face, one that depends heavily on uniform strokes and a rigid stance. In the font world, early American type founders called this a Gothic style, meaning not Greek or Roman but rather containing letterforms that have a relatively uniform stroke weight, that is, a block font (Figure 4). My goal with this style was to create the classic example of a Gothic font that is versatile enough to be used in a number of applications, particularly for display, while maintaining a somewhat narrow stance that would easily occupy layouts where space is at a premium. I also wanted it to be comfortably readable when viewed at a distance and at smaller sizes. That kind of tweaking, adjusting, and testing is a real challenge. It's also exactly the kind of challenge that CorelDRAW is designed to handle.

ROMAN
CLASSIC GOTHIC

Figure 4: Roman and classic Gothic fonts

Sometimes the best ideas start with pencil and paper. A hand sketch of the basic shapes that represent the American Captain font is the best way to illustrate how I want key letters to look (Figure 5).



Figure 5: Hand sketch of key letters

I'd like to keep the space around the letters and between the letters — the negative space — very minimal. Notice the top cross-stroke on the upper-case T, for example. It is very short, so that the space below the stroke, to the left and the right of the vertical stroke of the T, is also kept to a minimum. The actual strokes of the letters — the positive space — will be quite heavy.

With this concept on paper now, I can scan the sketch and import it into CorelDRAW as a JPG bitmap. I lock it into place with a simple right-click. In CorelDRAW, everything in the workspace — from scanned art to text — is an object. You can lock objects into place by using the **Lock object** command. First, select the

object. A bounding box with handles appears around the selected object. Right-click the object, and choose **Lock object** from the context menu. (You can also lock objects by clicking **Object ▶ Lock ▶ Lock object.**) The locks around the object indicate its stationary status. Locking the sketch prevents it from being accidentally selected and moved out of place.

 To quickly unlock an object, right-click and choose **Unlock object** from the context menu.

The next step is to fire up the drawing tools and create a few basic shapes directly on top of the scanned letters. I'll use these shapes to craft the vectors that will represent the individual letters of American Captain.

Creating the basic shapes

With the rough sketch locked down, it's time to draw the baseline shapes that will be used to build up the letterforms. All you're doing here is creating vectors, that is, giving digital life to your sketches and concepts. If you've worked with any version of CorelDRAW, digitizing will be familiar and easy.

Make sure you keep duplicates of the shapes you like. Set them aside in your work area so they can be used to build subsequent characters. The vectors you build in these early stages will serve as the basis for the forms and themes that will be repeated throughout the alphabet. So if you have something that's the right size and shape, preserve a copy for later use. Think of your workspace as a literal desktop. Projects finished and unfinished, even the scraps left over from your efforts, can freely clutter your desktop until you need them elsewhere, or until you're ready to discard them altogether.

For the CorelDRAW novice, it's going to take some practice to learn the tools, menus, and commands, but the procedures quickly become intuitive and natural as there are plenty of shortcut keys and alternative ways of accessing commands through menus, dockers, right-clicks, and so on. Once you discover them, it's easy to pick your favorites. The best way to learn more about the inner workings of CorelDRAW is by using it. Consult the help when you don't understand a feature, read all the instructional material you can find, and most importantly, open up a blank CorelDRAW page and start experimenting.

But before I get started, let me explain what a vector is.

Defining vectors

As mentioned earlier, a font is basically a collection of outlines, or vectors. A vector is a line or path that defines an object. A straight line is just a vector that has no curves. The vector's shape is defined by nodes, or control points, which in turn can be manipulated by control handles, the lengths and angles of which determine the final shape of the curve.

There are three types of nodes (Figure 6). Smooth nodes are designed to smooth the roundness of a curve. To add a sharp angle to a curve, use a cusp node. If you want to apply the same curve to both sides of a node, the node needs to be symmetrical. Moving a node's control handles changes the shape of the curve. With symmetrical nodes, pulling one handle causes the opposite handle to adjust itself identically. If a line is straight, it has no control handles. A straight line can be converted to a curve, and a curve can be converted to a straight line.

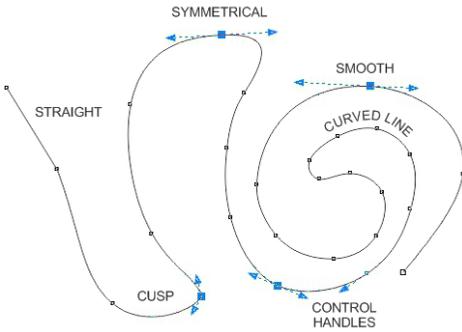


Figure 6: Different types of nodes

Nodes and control handles are adjusted by using the **Shape** tool. Alternatively, you can use the **Shape** tool to transform a line by simply dragging the line itself. To modify a node or a line, right-click it and choose an option from the context menu (Figure 7). The full set of node-editing tools is also available from the property bar.

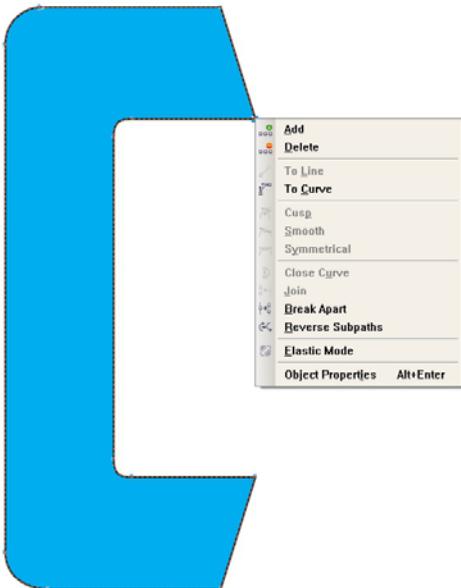


Figure 7: Right-click a curve or node to access the context menu.

When you view a vector onscreen in **Simple wireframe** mode (**View ▶ Simple**

wireframe), it resembles a pixelated line. Switch to the **Enhanced** view, and the rough and jagged line turns into a well-defined curve or vector with control points that are smooth or sharp, depending on the type of nodes along its path.

A vector that you can edit with the **Shape** tool is considered a graphic object. A graphic object must be a closed vector before it can be used as an outline in a font editor. That means that there can be no nodes that end in empty space (Figure 8). Every node has to be joined to another node, so that there is always a vector on either side of a node (Figure 9). Unjoined nodes can be joined by dragging one node on top of the other. This automatically snaps the nodes together. Or, while two nodes are selected, right-click and choose **Join** from the context menu.

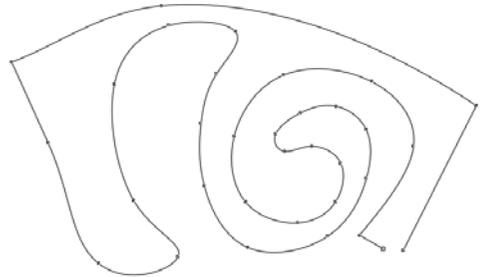


Figure 8: This is an open-path object, and it cannot accept a fill. The two nodes in the lower-right corner must be joined to close the path.



Figure 9: This is a closed-path object, which can accept a fill.

A quick way to create vectors is to use the **Rectangle** and **Ellipse** tools. You can easily rearrange these basic shapes, convert them to curves, and build simple letters, like an E or an H (Figure 10).



Figure 10: Rectangles as letterforms: outlines without fill (left), outlines with white fill (center), no outlines, red fill (right)

Rectangles and ellipses are basic shapes that cannot be edited node-and-line by using the **Shape** tool. You can modify them in other ways with the **Shape** tool, but to edit them as vectors, you must convert them to curves. You can convert each individual object by using the **Convert to curves** command (**Object ▶ Convert to curves**). Alternatively, you can weld rectangles and ellipses together to remove overlaps and create a single object that is also a graphic object (**Object ▶ Shaping ▶ Weld**) (Figure 11).

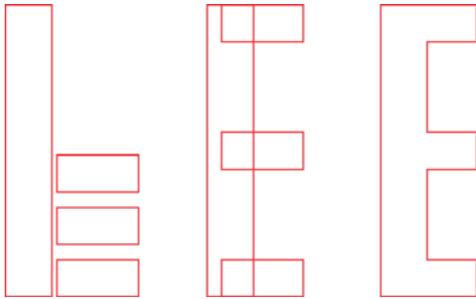


Figure 11: Is this four rectangles, or an upper-case E? Welding the rectangles together creates a single object.

Building the critical letters first

I'll start with the letter T because it is a critical letter for this font and will set the tone for the entire upper-case alphabet. First, I make sure the scanned image (Figure 12) is locked into place.

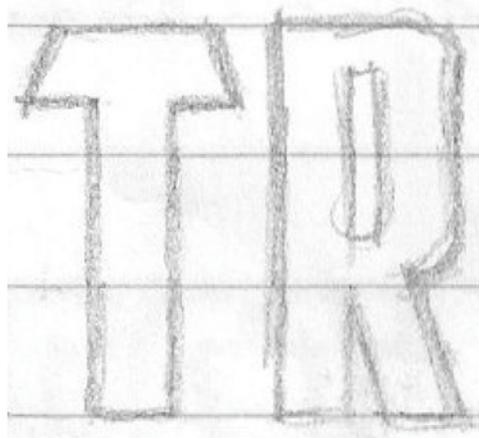


Figure 12: Scanned sketch

Now, I click the **Rectangle** tool from the toolbox and draw a rectangle from the bottom of the T to the top, roughly aligning it to the horizontal and vertical strokes of the scanned sketch (Figure 13a). I don't bother going all the way to the top of the T. You'll see why in a moment. I make a copy of the rectangle to use in the construction of the rest of the letterforms so they all have the same width. The height will be adjusted later.

Next, I draw a small rectangle that boxes in part of the top cross-stroke of the T (Figure 13b). I don't worry about precision at this point. I'll focus on precise placement of nodes, lines, and angles later. I convert the rectangle to curves. Using the **Shape** tool, I drag the nodes in the upper and lower left-hand corner of the object to create the distinct angle in the top stroke of the T (Figure 13c).

 To constrain the movement of a node or an object to a perfectly horizontal path, hold down **Ctrl** as you drag.

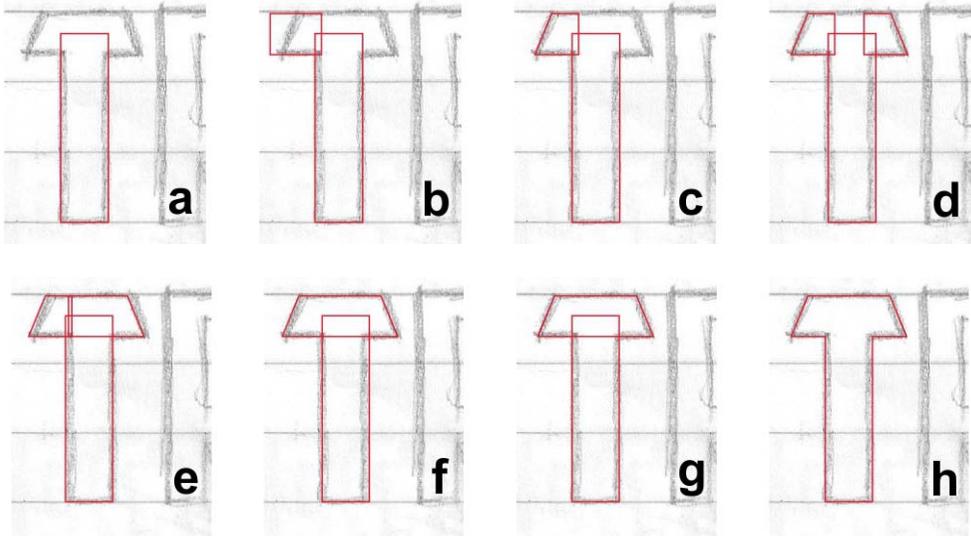


Figure 13: Building an upper-case T

I duplicate the cross-stroke, flip it horizontally, and drag it into position (Figure 13d). A quick way to flip and duplicate an object is to hold down **Ctrl**, select the object, grab a center control handle, pull it in the desired direction until you see a blue outline — mirror image of the object opposite your original — and then right-click. Release the left mouse button and the **Ctrl** key and you're done! You've duplicated the object and left the original behind. Anytime you right-click while moving an object, the object is duplicated while the original stays in place. Consult the help and menus for more options about duplicating objects.

When the duplicated cross-stroke is in position, I use the **Shape** tool to drag the nodes of one of the cross-strokes until they overlap the other cross-stroke (Figure 13e). I select both objects and weld them together (**Object ▶ Shaping ▶ Weld**) (Figure 13f). With both objects now welded into one, I use the **Shape** tool to select and delete any extraneous nodes. If any of the straight lines have

defaulted to curved lines, I click anywhere on the curve with the **Shape** tool, right-click, and choose **To line** from the context menu to convert the curve to a line. To delete any unnecessary nodes, I select them with the **Shape** tool and press **Delete**, or right-click and choose **Delete** from the context menu. Now I select both objects — the vertical stroke and the new horizontal cross-stroke — and align them vertically (**Object ▶ Align and distribute ▶ Align centers vertically**) (Figure 13g). To finish, I weld these objects together (Figure 13h).

 Double-clicking a node with the **Shape** tool deletes the node. Double-clicking on a line adds a node. Selecting a point on a line and pressing **Delete** deletes the line but leaves the vector open.

I notice that the T is 0.994" tall. I need to manually adjust the height of the duplicate rectangle I created earlier to make it 0.994" tall as well. I do this by typing values in the **Object size** boxes on the property bar. I click the **Lock**

ratio button before entering the new height to disable proportional scaling and preserve the width of the object. As long as the **Lock ratio** button is disabled, the only dimension that changes is the one I enter. Now both the width and the height of the duplicate rectangle can be used as a reference for subsequent letters.

As for the 0.994" measurement, it's obviously an odd size to work with, which is what happens when you throw things together without rulers, guidelines, or good planning. I'll show you an easy way to correct this later on.

Building the curved letters

So far I've been drawing only letters that have straight lines. Building curves into a vector to create the curved components of letters like the C and the S requires a different approach. In this case, I can round the corner of a rectangle by holding down **Ctrl** and dragging the corner with the **Shape** tool (Figure 14). No matter whether I drag vertically or horizontally, the corner becomes perfectly rounded, and I can easily modify the corner radius.

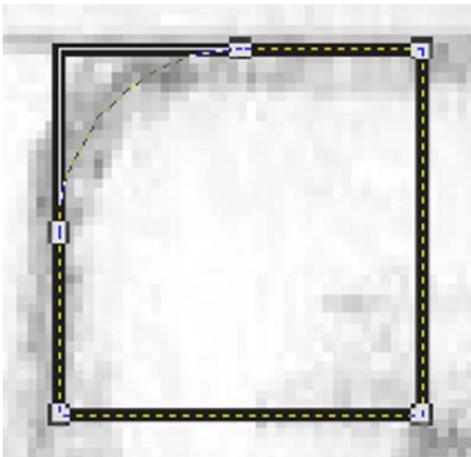


Figure 14: Rounding the corner of a square

Note that if you don't hold down **Ctrl** while dragging, every corner of the rectangle is rounded (Figure 15).

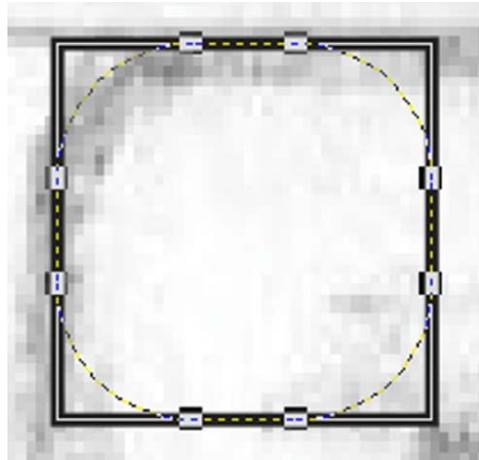


Figure 15: Rounding all corners of a square

Starting at the upper-left corner of the C, I draw a square with the **Rectangle** tool by holding down **Ctrl** as I drag (Figure 16a). Holding down **Ctrl** constrains the rectangle to a square. Next, I use the **Shape** tool to create a curve from the upper-left corner of the square, trying to mimic the shape of the letter C in the sketch (Figure 16b). After I create the rounded corner, I convert the square to curves. Next, I pull the right-side nodes of the square with the **Shape** tool until they match the shape of the top horizontal stroke of the C (Figure 16c).

At this time, I bring in the copy of the rectangle saved from the building of the T. I scale it vertically so that it rests within the confines of the top and bottom strokes of the C (Figure 16d).

I flip the top horizontal stroke vertically and duplicate it for the bottom horizontal stroke of the C. I drag the duplicate into position (Figure 16e), and then align all three pieces and weld them together (Figure 16f).

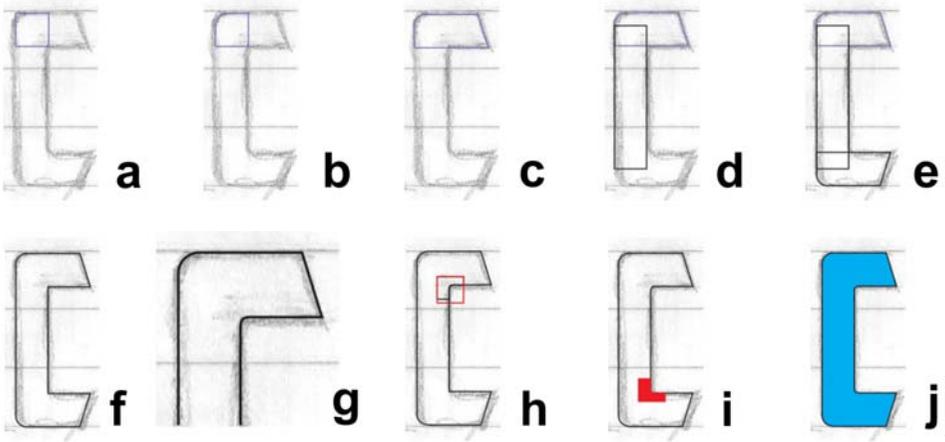


Figure 16: Building the upper-case C

For the inside arc of the C, I zoom in tight and build it by eye with the **Shape** tool (Figure 16g and Figure 17). First, I add three nodes and delete the middle corner node. Then, I convert the straight line to a curve and make the nodes

smooth to create a perfect transition from a straight line to a curved line. Finally, I build the arc with a few strategic tugs on the control handles.

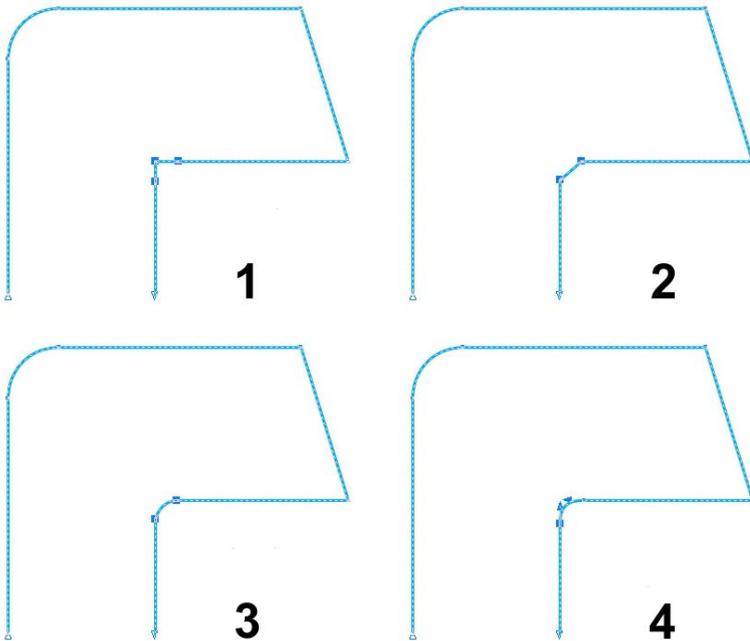


Figure 17: Creating the inside curve by 1) adding nodes, 2) deleting the middle node, 3) converting the straight line to a curve, and 4) fine-tuning the arc by using the control handles

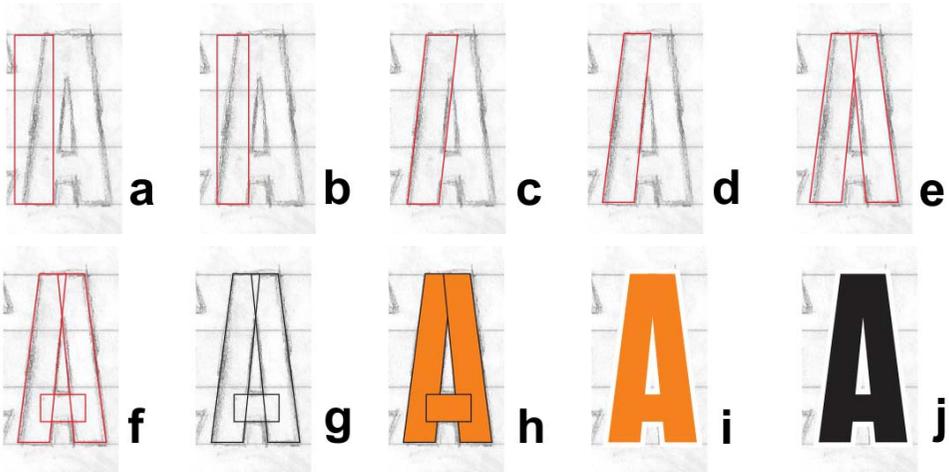


Figure 18: Building the upper-case A

 To quickly round a corner, click **Window ▶ Dockers ▶ Fillet/Scallop/Chamfer**. In the **Fillet/Scallop/Chamfer** docker, enable the **Fillet** option, and type a value in the **Radius** box. Select the corner node, and click **Apply**.

I could build the other inside curve of the C by eye too, or use the **Fillet** option, but I can also replicate the top inside corner by drawing a square, placing it on top of the corner, selecting both objects, and clicking **Object ▶ Shaping ▶ Intersect**. Intersecting allows me to cut out a copy of the inside arc (Figure 16h) and move it where it's needed so I can weld it into place (Figure 16i). After some tweaking and cleanup, the C is now finished (Figure 16j).

Creating the A

Before we move on, let me show you how to create a perfect upper-case A. First, I bring in the saved rectangle from the T (Figure 18a), and then I scale and drag it to match the width of the vertical stroke in the sketch (Figure 18b). I double-click the rectangle with the **Pick** tool, and when the skew handles appear, I skew the

top of the rectangle to the right (Figure 18c). I convert the rectangle to curves and make a quick node adjustment to line up the inside vector with the sketch (Figure 18d). Next, I flip and duplicate the object and carefully position the two pieces to form the basic shape of the A (Figure 18e). With a third, smaller rectangle, the cross-stroke of the A is finished (Figure 18f).

I press F12 to bring up the **Outline pen** dialog box and change the outline to black for contrast (Figure 18g). You can also right-click the black color swatch on the color palette to change the color of the outline. Then, I fill the shapes with color (Figure 18h). Before I weld the three objects, I remove the outline so I can see what the letter looks like (Figure 18i). If I don't like it, I can use my arrow keys to nudge the nodes around before I'm ready to weld. I keep the three objects in this A grouped together (**Object ▶ Group**) so they don't become separated. When I'm satisfied and I've tested the A against more letters of the alphabet as they're completed, I make a duplicate of the grouped A shapes for later

use, and then I ungroup and weld the shapes to produce a single object (Figure 18j).

Finishing the first seven letters

By the time I get to the S, I finally have some finished letterforms to work with. They aren't perfect, but I have seven now and I'll concentrate on them first before I go any further. To see how close I am to my goal, compare the letterforms in progress to the actual finished letters (Figure 19).



Figure 19: The letters in red are work in progress; the letters in black are the finished letterforms.

If you look closely, you'll find there's still some work left to do. For this article, I'm literally re-digitizing the American Captain font, so you can see the step-by-step process. It's impossible to make it look exactly the way it did the first time I digitized it, but I am close — plus I see some new things that I really like. Maybe there's a whole new font coming together here!

At this point, with your initial letters done, it's time to make sure that every line and curve and node is exactly where you want it to be. Try to keep the lines straight and the curves smooth and flowing with natural transitions — except where you don't want them smooth and flowing! And check your work. By zooming in close with the **Zoom** tool, I found an inside curve on the S that was slightly askew

(Figure 20). With the **Shape** tool, I moved the control handles and gave the curve a nice, smooth transition. The cleaner the vector, the better the letter!

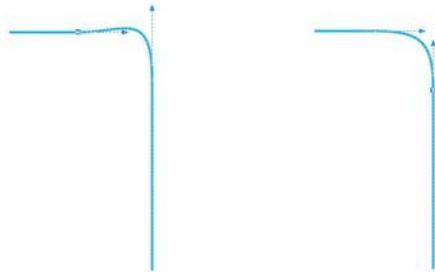


Figure 20: Adjusting the curve by moving the control handles

For more tools and commands that are available when a node or line is selected, check the property bar. And always consult the help when you're not sure how a tool or command is supposed to work.

 Switching from **Enhanced** view to **Simple wireframe** view (**View ▶ Simple wireframe**) can make it a lot easier to locate nodes and check the integrity of outlines.

And you're never limited to only one way of building a vector. When you put your favorite tools to use, the process will become even more natural and intuitive. By combining various techniques, you'll eventually find the ones that work best for you. Some of the images in the following examples (Figure 21) show that there is more than one way to build an outline. The letters in red are the final products.

TRCRSEAD AA
CCCCC CC DDDDTTC
DTC

TRDRSEAD
DDDBRRR RDTCA

TRC CSEAD
C C S S S S ERDTCA

Figure 21: Creating new letters from existing letters

Testing the letters in words

Even if you didn't start out with a word, create one now. The letters in my sketch offer a limited number of combinations to form a real word. Leaving the T aside for a moment, I get a first look at the new letterforms in action by spelling out SACRED (Figure 22). This is a particularly nice letter combination. I get to see how the negative space inside the loops of the S interact with the negative space around the A.



Figure 22: Testing the letters in different real-word combinations

Real words help you establish how the font looks in familiar arrangements. The letters don't look too bad, but they are still pretty rough. It's time to nudge the nodes in some of the angles and alter them slightly. With the **Shape** tool, I select each node I want to move and use the arrow keys to nudge it horizontally or vertically.

 To set the nudge distance, click the **Pick** tool, click in an empty area of the page to deselect all objects, and type a value in the **Nudge distance** box on the property bar.

I set the nudge distance to 0.005", just enough to fine-tune the shape of the letters without making major changes. When adjusting the angles on the S, I count the number of nudges so I can adjust the angles on the C, E, and T by the same amount. This way, the angles in all the letters that share this angle will be equal. If

you make a mistake or lose track of the nudges, you can always use the **Undo** command (**Edit ▶ Undo**) to go back and try again. Or you can manually manipulate the angle to match the angle of a letter that you like.

Looking at the letters, I see that the R definitely needs improvement. It is a bit narrow and has to be stretched. I select it and pull the left or right middle control handle along the horizontal axis until the overall width of the R coincides with the width of the other letters. The scale factor is about 108%. The width of the vertical strokes changes a little, but not enough to be noticed.

After I'm done with these minor changes, I can rearrange the letters to test the word SCARED. I like it. Now I add the T. CRATED is a good test word, but I'd like to include the S as well, so I try DESECRATED. From this new word I see that I need to adjust the S slightly with a few more node nudges, and I also tweak the cross-stroke of the T to make it a little wider.

For now, I'm very satisfied with the results.

Before I continue, I'd like to apply some graphic elements to my first words. It's wise to consider how things like drop shadows, fills, outlines, and so on affect the appearance of the letters (Figure 23).

You may want to make various adjustments when you see the letters in action. Try to finalize the weights and strokes during these early stages of creation, when the overall look is beginning to take shape. Making these adjustments now will allow you to carry them over to the rest of the font later. Otherwise, you'll find yourself trying to modify the whole alphabet at the end of the process.



Figure 23: Checking the overall look of the letters by adding graphic elements

Be careful while you're adjusting lines and curves, nudging nodes, and scaling objects. Make sure you maintain the integrity of the angles and shapes you like. Keep rearranging your letters and rechecking how they look in different combinations. Now is a good time to introduce a radically different shape like the V or the W into the equation.

Creating the V

Remember the duplicate of the grouped A shapes that I set aside earlier? Now I ungroup the objects, remove the cross-stroke, weld the remaining two pieces together, rotate the new object 180°, and I've got a V (Figure 24).



Figure 24: Creating a V from the A

To rotate the object 180°, double-click it with the **Pick** tool to display the rotation handles, hold down **Ctrl**, and drag a corner handle (Figure 25). Holding down **Ctrl** constrains the rotation to 15-degree increments, making it easier to control the angle of rotation until you reach 180°.

Another way to rotate an object 180° is to select the object and click the **Mirror vertically** button on the property bar.

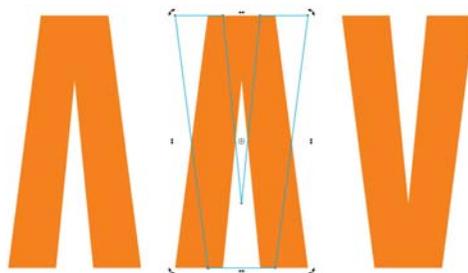


Figure 25: Rotating the object

Now I want to see how the V looks, especially next to the A. I feel it needs some fine-tuning, so I nudge the nodes around on the top, bottom and middle of the letter. The most drastic change is on top, where I widen its "flare" (Figure 26). The blue-outline V in the example is now my working version. There is considerable improvement from the initial V. But is it right yet?



Figure 26: Making adjustments to widen the flare of the V

It's probably fine, but I'm not sure. It looks too skinny. I want to make it slightly bigger without nudging any nodes, so I add an outside contour by clicking the **Contour** tool in the toolbox and dragging away from the center of the object (Figure 27).

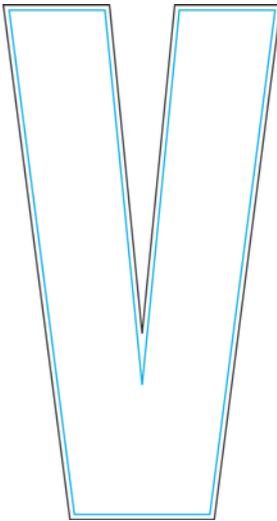


Figure 27: Adding a contour to widen the V

Of course applying a contour will affect the height, but I can change the height by specifying values in the **Object size** boxes on the property bar. This imprecise adjustment

would likely affect any letters that have cross-strokes and angles, like an E or a T. But on a V, who can tell? Besides, I can already see that once I separate the contour from the original outline, this new V isn't going to work. It's just too bulky (Figure 28). The working version — the V that I nudged the nodes on to widen the flare — is “working” just fine.



Figure 28: The top V is too bulky; the bottom V is perfect.

Scaling the letters

Keep rearranging your letters and checking them often. If your creative process is hasty and chaotic like mine, some of your measurements may be off by now and may need readjusting.

For instance, to change the odd height of my letters — they are all currently 0.994” tall — and make them 1.0” tall, I set up 1-inch guidelines and align the letters to them. One way to set up guidelines is by using the **Options** dialog box (**Tools ▶ Options**). However, I normally just use the **Pick** tool to drag a couple of guidelines out of the rulers and place them at a point in my workspace that is not too cluttered. Then I double-click one of the guidelines to display the **Options** dialog box. Now I can type values for each

guideline that will give me the space I want between them.

Next, I click **Snap to** on the standard toolbar and enable the **Guidelines** check box. I duplicate one of the letters (the A will work), drag it between the guidelines until it snaps into place on the bottom guideline, and scale it proportionally with a corner handle until it snaps to the top 1-inch guideline (Figure 29). When scaled, the A snaps to the 1-inch guideline and becomes 1-inch tall.

Now I select the remaining letters and press **Ctrl + R** to repeat the action. The **Repeat** command applies the scaling performed on the A to the remaining letters and makes them all 1-inch tall, the same height as the A.

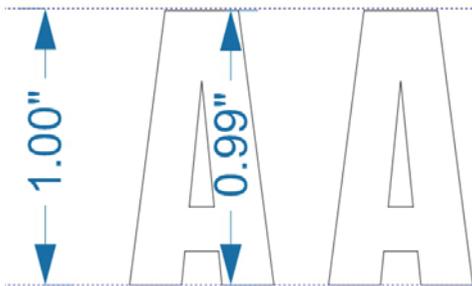


Figure 29: Scaling the A to fit between the guidelines

Of course, the assumption here is that I had maintained the odd 0.994" height of the letters as I created them. If a letter had varied in height, it would not be 1-inch tall now. If you find a letter that isn't exactly the size it should be, correct it now. When an object is selected, you can check its dimensions in the **Object size** boxes on the property bar.

Fixing the cross-stroke of the T

And what did I say about messy creative process? I notice that the cross-stroke of the T is a little wider than the ones in the E, C, and S (Figure 30). Time to fix it.



Figure 30: The cross-stroke of the T is wider than that of the E.

I drag the E over the T, change the **Snap to** selection to **Objects**, select the nodes on the T that need to be moved, hold down **Ctrl** and drag upwards. I zoom in tight before making precise adjustments like these. It's easier to see what I'm doing (Figure 31). I select the nodes first and then zoom in on one of the nodes before I drag it. The other nodes will follow suit, even if I can't see them.



Figure 31: Zooming in and adjusting the cross-stroke by moving nodes

But guess what? Now the slanting angle on the cross-stroke is off. I drag the T so its right-side cross-stroke lines up with the cross-stroke of the E (Figure 32) and nudge the bottom node until the entire angle is right again. I repeat the same number of nudges on the left side of the T. Angles fixed! There's so much work and so many details to consider! And you thought making fonts was a breeze! But I tried to warn you — it's all about vectors. And nodes. Lots of nodes.



Figure 32: Aligning the right cross-stroke of the T with the E. The black color along the right edge shows that the T needs to be adjusted.

Actually the angle was off by so little that I could have probably just left it and no one would have noticed. In this case, I zoomed in over 3,000% and the difference in the angles was still negligible. But why not fix it if I can? I set the nudge distance to 0.0001" and it took 12 nudges to get the node back to the right position. Before I forget, I switch back to a more realistic 0.005" nudge distance.

Creating the rest of the alphabet from the initial letters

Now I have to create the remaining letters of the alphabet. This Gothic font relies on a

number of repeating shapes to form the complete character set, so you can practically work assembly-line style. Since all the letters must be the same height, I'm going to build them by using the guidelines. I enable the **Snap to ▶ Guidelines** option and lock the guidelines (right-click each guideline and choose **Lock object**), so I don't grab one by mistake and move it out of place.

I start by duplicating the letter E, which can be used to quickly build the letter F (Figure 33). With a few simple clicks to delete some nodes and drag the remaining node to the guideline, the E becomes an F. Next, I duplicate the F, delete a few more nodes, snap the last one to the guideline, and I have an I. But what if I leave the center horizontal stroke on the F, straighten its top and slide it next to the I? Now I have an H. I adjust the space and weld it together. The D will be the pattern for the space between the two vertical strokes of the H.

Next, I slice a D in half vertically by drawing a rectangle and positioning it so that it overlaps the left half of the D. I select the rectangle, hold down **Shift**, and then select the D (the order of selecting the objects determines which of the two objects is cut). I click **Object ▶ Shaping ▶ Trim** to slice the D with the rectangle, and I remove the left side of the D. Now I create a mirrored duplicate of the remaining right half of the D and weld it back to itself for an O. I drag the nodes to get the middle space the same as the space inside the D. As I work, I keep the D in the background for reference, switching to simple wireframe view for greater precision.

E F F F F I I I I D D H

J O O O

N H O W H O O Q O U

O U R R R P P B B

W W W Y

P V W X X W W W Y 3 1 2 2 Z Z

Figure 33: Creating more letters

A rectangle placed at the bottom center of an O turns it into a Q. I put two Vs together to form a W (with lots of heavy node tweaking, of course), then slice and dice an R to make the P and B. The P needs some curve adjustments to look right. The B is even more complicated than the P, but careful tweaking of the nodes and the control handles will make that little indent look just right.

Line up the new letters and make adjustments where necessary. There's a special tool for this. It's called your eyes. Train your eyes to see type. Train them to understand and visualize not only the positive space occupied by the letters, but the negative space around the letters too. Negative space is an important element in design and composition. An example of effective use of negative space are layouts with reversed letters on black backgrounds. To prove the importance of negative space, let's take a look at three identical red shapes complemented by a quick drop shadow effect (Figure 34).



Figure 34: Three shapes

Cool, yes? But what do you really see? When I change the view in the next example, you see that it's a carve-out of the negative space between the W and the S, rotated 180° (Figure 35).



Figure 35: The three shapes are made from the negative space between the W and the S.

Fooled? No? At least not for long. Did you see the objects as positives or negatives? Or did you immediately see them for what they were? Did you find them interesting? That's the goal of negative space — to add interest to your designs. Granted, a Gothic style like American Captain limits the possibilities because the letterforms are so simple, but where you can, try to add interest. When you're working on the positive space, never lose sight of the negative space.

Building a font as described in this article is nothing more than simple extrapolation — using existing shapes and forms as starting points to create the remaining forms. Employing the vector-editing techniques of CorelDRAW, from welding and trimming to duplicating and mirroring, will help you create all the pieces you need to make the rest of the alphabet.

Remember to check your work often as you go. Overlooked mistakes can carry over from one letter to the next until your whole alphabet is riddled with glitches that should have been corrected at the start. Look for errors such as lines that are supposed to be straight but aren't, smooth transitions versus cusps, angles that are off, and so on. Remember the inside curve of the S that was slightly askew? Those are the kinds of mistakes you should be looking for so they don't infiltrate the rest of your font by way of the extrapolation process.

Tweaking the C

Unfortunately, sometimes I don't follow my own advice too well. Deep into the process (26 letters later, to be exact), I discovered a flaw that should have been caught while I was digitizing the first seven letters from my sketch. I noticed that the angles on the lower horizontal cross-strokes of the C and E don't match the diagonal slope of the A (Figure 36).



Figure 36: Adjusting the lower cross-stroke of the C to match the slope of the A

When I move the C to the left of the A, the discrepancy is obvious. For contrast, I change the fill of the C to black. Holding down **Ctrl**, I drag the C horizontally until it just touches the left-side diagonal of the A. I switch to wireframe view and use the **Zoom** tool to zoom in tight on the area where the two letters touch. This magnified view helps me accomplish these close-in maneuvers with ease. I set the nudge distance very low (0.001") and use the arrow keys to nudge the C into place (12 nudges in this case). I get it as close as I can, but I'm not overly concerned with absolute pixel-on-pixel accuracy.

The slant on the C now matches the angle of the slope on the A. I repeat the same number of nudges for the respective nodes on the E, L, and Z. The awkward negative space between the letters and the A is gone.

I also realize that the corners of the C aren't as rounded as I need them to be. They should look more like the corners of the D and the O. The letters that need adjustment are the C, E, L, and Z. If I round the corner of the C, I also

have to round one or both corners of the G. Even the S will have to be tweaked again.

With the **Shape** tool, working on the lower-left side of the C, I begin by moving the nodes around to manually create the shape I want to see (Figure 37). When a curve starts to straighten out too much, I use the control handles to pull it back out again. All the adjustments are visual. I'm not going for precise measurements. I'm just trying to make it look right.

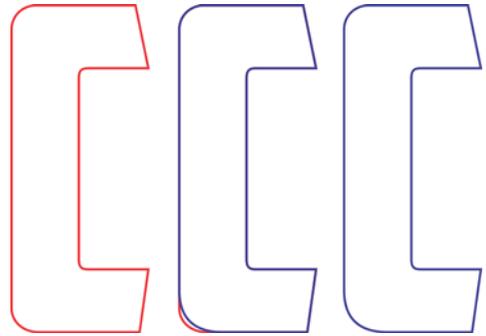


Figure 37: Correcting the rounded corner of the C

Once I'm finished with the corner of the bottom cross-stroke, I need to transfer the changes to the top cross-stroke as well. You can always use the **Fillet** option again, making sure that you enter the same value for the corner radius that you used before, but here are a few other tricks and tools you can use.

I duplicate the C and select all nodes by dragging around them with the **Shape** tool (Figure 38). I right-click to bring up the node-editing context menu and choose **Cusp** to turn all nodes into cusps. Next, holding down **Shift**, I click the two nodes at the bottom left of the C to deselect them. These two nodes represent the newly built arc. The other nodes are not needed and should remain selected. I release the **Shift** key. Now, with all the remaining nodes still selected, I press **Delete**. I'm left with

an odd-looking shape, but you'll notice that the arc I built is still intact (Figure 39).

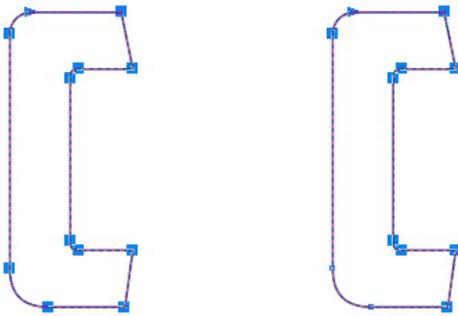


Figure 38: Selecting all nodes of the C and then deselecting the two nodes in the lower-left corner

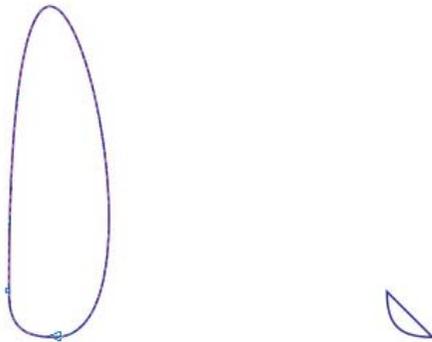


Figure 39: The two nodes in the lower-left corner and the arc between them remain intact.

Because the nodes that define the arc are cusps, the arc remains intact even if you modify or remove that odd curve that loops around either side of the arc. The next step is to delete the curve. I simply select any point on the curve, right-click, and choose **To line** from the context menu. The curve is gone. A straight line remains. The arc is intact.

I flip the arc vertically and move it to the top of the C (Figure 40). I select the straight line on the arc object and press **Delete**. The arc is now an open curve. I could have deleted the curve initially instead of making it a line, but for the sake of example, I decided to show you how to

straighten the odd curve first, instead of deleting it right away. Next, I select any point on the original top rounded corner of the C, click it with the **Shape** tool, and delete it too. The C is now an open curve. I select both objects and align them by clicking **Object ▶ Align and distribute ▶ Align left**, and then **Object ▶ Align and distribute ▶ Align top**.

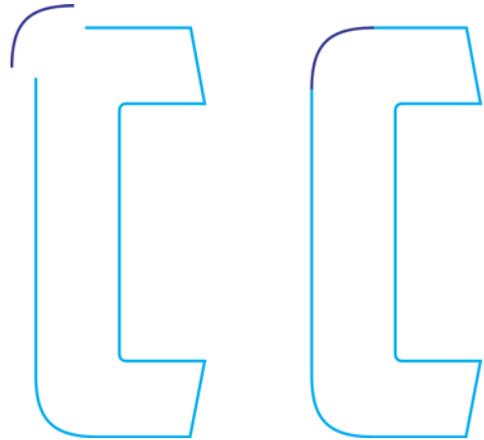


Figure 40: Moving the arc to the top of the C

I combine the two objects into a single curve (**Object ▶ Combine**, or **Ctrl + L**) and move the nodes on the straight lines of the original C shape until they snap to the nodes of the new arc. I'm careful not to move the nodes on the arc and change its shape. When the nodes from the two vectors have properly snapped to each other, they will automatically join. When both are joined, the object turns from an open curve to a closed curve. The letter is finished once it is a closed curve.

Now I can make similar corrections to the G, but only to the lower-left corner. The upper-left corner will remain unchanged. I even remembered to save a copy of the arc I created, so it's a simple matter to join it to the bottom of the G the same I way I joined it to the top of the C. The S will need some attention too.

YMJBAETDUZICQVFWHNRPGKLOXS AMERICAN CAPTAIN

Figure 41: The finished alphabet

When you're finally finished, line up the 26 letters to see how they compare to your initial concept (Figure 41). And here's a useful tip: Don't arrange all letters in alphabetical order. Any font that is arranged in alphabetical order looks good to your eye. Mix it up. Separate out the V, W, X, Y and Z. Put them anywhere but next to each other, or you'll never be able to tell if they look right.

If you feel intimidated by the idea of creating fonts, it's time to start learning all you can about type design. That goes for the veteran users too because there's something else to keep in mind if designing fonts is going to be an ongoing venture. This article deals only with 26 characters. A complete font set requires a minimum of 250 characters. Not to scare you or anything, but when American Captain was finally done, it took about a year to complete, encompassed multiple languages, included five supplemental fonts, and tipped the character count at just over 1,600. On top of that are hundreds of character replacement options as OpenType features.

Et tu, AMCAP? From simple to complex shapes

So far I've been building relatively simple characters that require only basic construction techniques. But there are times when certain shapes can't be easily extrapolated from existing forms or welded together out of simple squares and rectangles. While it might be easy to build an upper-case H directly on the

screen without a sketch, a truly complex character — a classic ampersand, for example — presents a much bigger challenge (Figure 42).

The ampersand that I envision requires a more fully realized hand-rendered drawing. In my initial sketches of the first seven letters, the shapes were rough because I knew they would actually be fairly easy to render. When it comes to ampersands, however, even one built in the straightforward style of American Captain can present unexpected difficulties. Et tu, AMCAP? In Latin, "et" is the conjunction "and," so an ampersand is actually a highly stylized ligature of the letters e-t (or E-t).

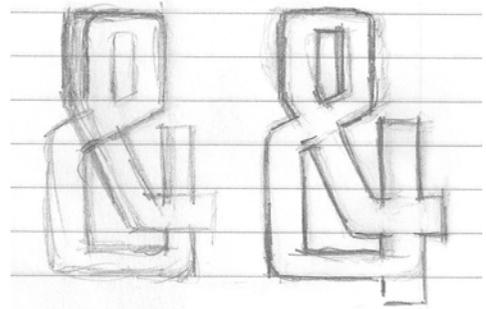


Figure 42: Rough sketches of the ampersand

The final artwork for my ampersand concept isn't perfect and doesn't have to be. But it is more precise than my initial letters. Working out the details of this particular character on paper will definitely facilitate the digitizing process. I've even broken it apart into individual shapes in case I decide to digitize the pieces one at a time and then weld them

together. This will also give me a better understanding of the overall form I'm trying to replicate.

But I have another idea first.

If I take the time to examine the shapes separately and borrow some of the necessary elements from the finished letters, it might actually be possible to digitize this ampersand without even scanning it into CorelDRAW. By simply familiarizing myself with the various structures that comprise the overall shape, I can construct and assemble this ampersand visually.

Viewed individually, the parts are actually pretty basic. They're simply vectors with a few extra angles and curves. And that means a little extra work. It's just a matter of building up the shapes, assembling them, and making final adjustments.

The first attempt looks a little different than expected. It took about 13 steps to get the shape I wanted, with a total work time of about five minutes (Figure 43). Plus I built it by using only three main shapes instead of the planned four (Figure 44). Plans often change

once you apply an idea to the constraints of the actual letterforms.

Working without the scanned sketch is tougher than I thought. If I want to stay true to the letterforms I already have (I'm using the P as a baseline character against which to check my work), my ampersand concept might have to remain just that — a concept. Things don't always work, but it's nice that I can quickly try out an idea and decide whether to keep going or abandon it altogether.

I decide to soldier on, even though I realize I'll have to commit a lot of time to the final adjustments of the stance, width, stroke, and so on. It's the only way to get this character right. The adjustments are numerous and time-consuming, so I want to avoid as much as possible the cumbersome task of moving objects around with the mouse. Instead, I use selective node nudging. Working at low nudge distances, you can adjust and tweak a vector with incredible precision. You can nudge the lines, the curves, the nodes, and the control handles of a vector. This technique can be applied especially well if you work on top of scanned and highly accurate artwork.

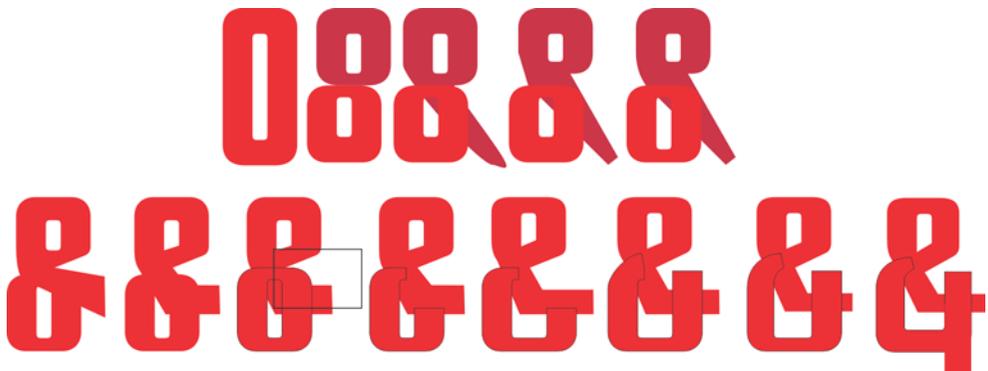


Figure 43: Building the ampersand from several separate shapes

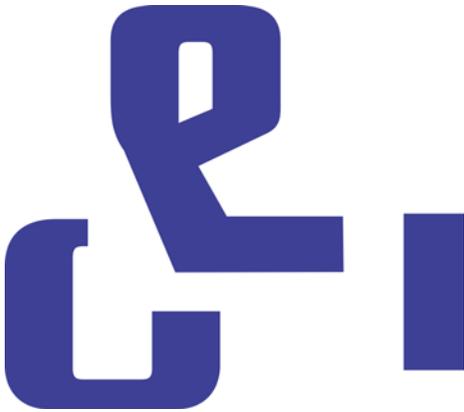


Figure 44: The three main shapes of the ampersand

Now it's time to re-digitize the ampersand by using the artwork as a guideline. This time I create the ampersand as one object, building straight lines and curves and tweaking nodes until everything is just right. It goes very fast, much faster than trying to build it by eye. But is it going to be any better?

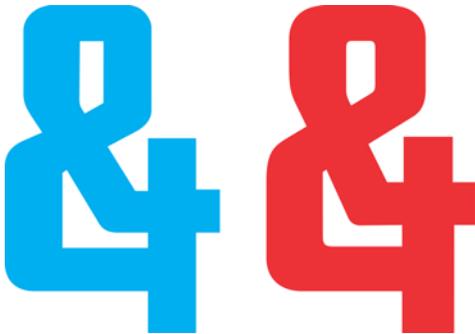


Figure 45: The two ampersands: The new one (blue) and the old one (red)

There is a notable difference between the two versions (Figure 45). I'll definitely save one of them for use as an OpenType alternate character. When I built the actual main American Captain ampersand, the one that you enter directly from the keyboard, I settled on another classical style. It's less extravagant, but it was difficult to create too. For my new

alternate ampersand, I decide to keep the first one, the one I built by eye.

Accessing font characters in CorelDRAW

I can easily access the new ampersand. CorelDRAW now has the ability to apply character replacement options from fonts that include OpenType features or a user-defined range of alternate characters. There are three ways to find the hidden extras in a font. First, the **Text properties** docker (**Text ▶ Text properties**) lets you explore available OpenType features from categories like Stylistic Alternates and Standard Ligatures. Second, you can highlight a string of text — or even a single character — to access a dropdown menu that displays replacement options specific to your selection. To do that, select the text with the **Text** tool, and make sure that the **Interactive OpenType** button on the property bar is enabled. If the selected text has any OpenType features, an arrow appears below the text. Click the arrow to see the available features (Figure 46).



Figure 46: Displaying the available OpenType features for selected text

Third, position the cursor where you want to insert a character in a line of text, and open the **Insert character** docker (**Text ▶ Insert**

character), which is basically a font character map for all standard characters, excluding OpenType features. From the **Character filter** drop-down list, enable the **Entire font** check box. Scroll through the map, and double-click the character that you want to insert. If an alternate character hasn't been written into the font as an OpenType feature, it is available only from the **Insert character** docker.

As a font designer, you can include a practically unlimited number of alternate characters in your font, with no requirement to make them available as OpenType features. It is necessary, however, to apply private-use Unicode designations to any alternate character; otherwise, the font will not function as expected. For more information, consult your font editor and any other source of typographic knowledge you can find.

Time for more testing

Now is a good time to put everything you have created so far to a practical test. Before you start importing vectors into a font editor, it's a good idea to use your new letterforms in actual words and see what works and what doesn't. Are any of the letters too wide? Too narrow? Are the strokes and weights right? Do the inside spaces need to be adjusted? The list of likely problems runs long, but deciding what looks right or wrong will be a judgment call. If you're not sure whether something is right or not, find a similar font and see how its designers dealt with the issue. Maybe their solution will work for you.

CorelDRAW has all the tools you need for laying out the letters into words and phrases to create page layouts, full-color arrangements, logos, and so on. It's a lot of work to put your letterforms through these tests, and I know you're eager to get your letters out the door,

but devote some effort to word testing first. This will really give you a feel for what works and what doesn't. And fixes may be easier now in CorelDRAW rather than later in a font editor.

Certain words and word combinations highlight problem areas really well. One of my favorites is FOES TOAST GHOST in all caps (Figure 47). It brings into the spotlight the difficulties inherent in words that use dissimilar and contrasting shapes. Other favorites are FLAG and CHEIVAN. For the CHEIVAN test, alternate the W, Y, and X in place of the V. What unique combinations can you think of? Here's a classic example: WAVE. Or maybe you prefer pangrams: QUICK JOLTING ZEPHYRS VEX FAMED BOW, or JACKDAWS LOVE MY BIG SPHINX MADE OF QUARTZ. The JACKDAWS pangram is particularly useful as its letters can sometimes (depending on the font) create a distinctly noticeable arrangement of the dissimilar forms I was talking about.

Try to put together combinations that are problematic for the specific style of your font. For instance, American Captain has a very small horizontal stroke on the upper-case T, so the question was obvious. Will this look good in real-world applications? Is it a sound concept? It's fine in the middle of a word, but is it too small at the beginning of a word? Only time and testing will decide.

I ultimately decided to keep the small top on the T, even if it does look a little bit weak at the beginning of a word. It is actually a hidden advantage. A small top stroke eliminates the excess negative space typically associated with an upper-case T.

YMJBAETDUZICQVFWHNRPGKLOXS

FOES TOAST GHOST CHEIVAN FLAG WAVE

FOES TOAST GHOST CHEIVAN FLAG WAVE

QUICK JOLTING ZEPHYRS VEX THE FAMED BOWS

COREL X6 JACKDAWS LOVE MY BIG SPHINX OF QUARTZ

013456789 AMERICAN CAPTAIN by the Fontroly

YMJBAETDUZICQVFWHNRPGKLOXS

Figure 47: Word tests

Defining the space between the letters

Now is also a great time to decide how much space you want between the letters, so you can apply it to the finished font in the font editor. For American Captain, the letter spacing is fairly tight. I kept everything close on purpose. This was something I wanted from the beginning, and once I saw it taking shape onscreen, I knew I would also need a plan to set up the letter spacing equally tight in my font editor.

Luckily it's easy to visually adjust the letter spacing in a font editor, with hardly any need for a lot of complicated measurements. However, there is a way to export the exact letter-spacing measurement directly from CorelDRAW to the editor.

Draw a square between two precisely spaced letters, making sure the square is touching the side of both letters (Figure 48). The square can now be sent with the letters into the font editor. The square's width divided in half will

help you determine the left and right side-bearing distance for these letters. In this example I'm showing an M and an N. I could even scale the square to 50% prior to exporting, noting that its new width will provide the exact side-bearing distance of these two letters, and all other letters that share the left-side and right-side characteristics of the M and N.

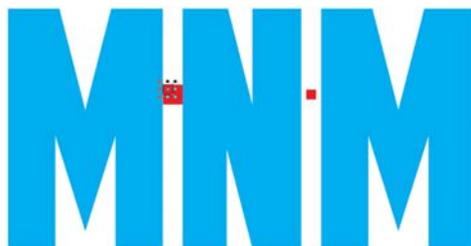


Figure 48: Creating a square and scaling it to 50%

For example, the left side of the E would have the same side-bearing value as the M and the N. The right side of the E obviously needs to have an entirely different value to look good.

Exporting the vectors

When you have adjusted everything to your satisfaction, it's time to export the letters. They'll be a typeface very soon, but there's a good chance that you'll revisit some of them many times in your ongoing effort to perfect the new font. While it's easy to make basic corrections in the font editor itself, the flexibility and intuitiveness of the CorelDRAW workspace, combined with its node-editing arsenal, give it a much greater capacity to handle the more complex final corrections.

When the vectors are finally imported into a font editor, they have to occupy about 75% of a grid space that is about 1000 total vertical units. A recommended value for the vertical height of an exported vector is between 700 and 800 units, or the equivalent in inches — about 7.0 to 8.0 inches before it leaves CorelDRAW. Typically, you scale the letters proportionally to the X-height since the height of the upper-case X is the standard for determining the final height of all upper-case letters. This height should be anywhere between 7.0 and 8.0 inches. This ensures the right scale for the letters to fit in their cells in the font editor and still provides the right amount of space between lines of text when you actually start typing the letters onscreen. Guidelines can help with the scaling. Make sure you scale all letters proportionally so they're uniform.

To align the letters so they arrive in the new environment with the proper orientation, set a vertical guideline to 0, and a horizontal guideline to 0 (Figure 49). Snap the vectors to the horizontal guideline, and from left to right, snap them to the vertical guideline. It's basically the same as aligning everything to the bottom left-hand corner of the page background. And it doesn't matter that the

letters run off the page, because they certainly will.

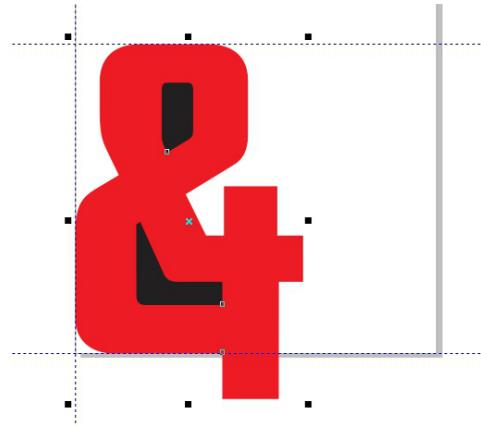


Figure 49: Aligning characters by using guidelines

The horizontal guideline is the baseline of the entire alphabet. The bottom of your letters (the upper-case M, X, and H are usually the default examples) should rest on the horizontal guideline.

If the vectors don't arrive in the editor exactly where you want on your first attempt, don't worry. You'll quickly learn to make adjustments in CorelDRAW so that all characters import into place, and you can also adjust their positions in the editor.

Before exporting a vector, make sure that it is a closed object and that it is broken apart in the case of letters that are more than one object (such as A, D, or O). For best results, remove the outlines (choose **None** from the **Outline width** drop-down list on the property bar), and apply a fill to all objects. Do not combine letters into a single object. Letters can be grouped, but not combined.

Consult the help in your font editor to determine what types of files can be imported. Typically, you'll be exporting vectors as an Adobe Illustrator AI file. Older versions of AI

files seem to work best for Fontlab. All options are available in the **Compatibility** drop-down list in the **Adobe Illustrator export** dialog box.

Before you open the **Export** dialog box, make sure you select only the objects that you want to export, and then make sure you check the **Selected only** box. Otherwise you risk making the mistake of inadvertently exporting everything in your workspace, which could lead to some catastrophic results when you try to import the file into your font editor. Failure to heed this warning will probably require a restart of your font editor and possibly CorelDRAW. In extreme cases, you may need to restart your computer. You've been warned!

It isn't necessary to export the entire alphabet all at once. While it doesn't actually hurt to export 26 letters in one file, I like to export only five or six at a time. This keeps unnecessary clutter out of the editor's workspace. Most font editors aren't going to be as forgiving of a messy user as CorelDRAW is when it comes to stacking up piles of vectors in the work environment. Strive to keep things at visually and technically manageable levels.

Vector tools in font editors are about precision placement of nodes and vectors — you'll see this glaring reality when your vectors arrive from CorelDRAW. If your vectors are not clean, the curves and the transitions are not smooth, or straight lines are slightly off, it will be obvious in your font editor.

The reason for this is that CorelDRAW is geared more toward creativity, not grid-like precision. But CorelDRAW can be just as precise if you pay attention to what you're doing. And the more precise the CorelDRAW vector is, the better. Cleaning up your work before you export it means less work in your font editor.

Conclusion

As I said at the beginning, this article deals with designing fonts using nothing but a scanner, a mouse, and CorelDRAW X7. The methods I've described are classic vector-generating techniques, but just as there is more than one way to build a vector, there is also more than one way to build a font. And more than one kind of tool to do it with. Even CorelDRAW is complemented by CorelTRACE, which lets you take your artwork from hand sketch to vector in a bit more of a hands-off way. And for many designers, pen tablets have replaced the venerable mouse entirely. So this article is not only a very basic introduction to the world of font making, but it also uses the simplest of hardware options, along with a very straightforward software means of generating vectors.

However, the methods I've described aren't exactly antiquated. Not yet, anyway. They're still very much an industry standard. And once you know what you're doing, it's a reasonably fast process. Just don't go overboard and try to put a thousand characters into your first font. Be satisfied to get the first 26 letters done and looking right. Work hard and tweak often, testing and retesting the letters until everything looks exactly the way you want it to look.

Of course, the ultimate test is installing the finished font and typing it into the CorelDRAW workspace. There you can easily assemble phrases and words and paragraphs and anything else text-related for which you'd like to use the new font.

 For best results, install the font as an OpenType postscript. This format retains a vector's original formatting information,

such as curve shape and node placement, and is therefore best suited for editing.



To quickly modify a letter typed in CorelDRAW as artistic text, first convert it to curves (**Object ▶ Convert to curves**, or **Ctrl + Q**).

CorelDRAW is ideal for quickly and intuitively running a font through its paces. It's also good for locating any overlooked flaws hiding amongst the letterforms. So those vectors you

were ready to declare finished? Sometimes a font project is never truly finished. There's always just one more idea, one more thing to correct, one more addition that will really bring it all to life.

Don't be surprised if you find yourself revisiting a typeface you thought long-completed. You never know when you might just decide to drop a funky ampersand into the mix — simply because you can.