Dubberly Design Office — March 2016

A Guide to Mapping Software Applications Quickly and Efficiently

Ryan Reposar — Designer — ryan@dubberly.com

Software applications suffer from the 'keyhole effect'—by looking through the viewport, you can only see one screen at a time. That means its difficult to immediately understand the scale and scope of an application, and it can be hard to know if you've visited every screen or seen every case. A way to bypass this is to create an application flow map; a single document which captures and presents every screen within an application and shows how they're connected.

We first developed these at Apple to understand HyperCard Stacks.

Our office has found these documents to be extremely useful for clients. They provide a view of both the forest and the trees, a common model for people across the organization to make sure they are on the same page.

Naturally, I've refined my technique over the years and this document attempts to capture all the details. Finally, I'd like to thank Greg Baker, who created much of what is presented here. I'm simply passing on what I've learned.

— Ryan Reposar

Step 1: Use the app

The first in creating an application flowmap is to get acquainted with the app. Use the app and try to be as thorough as possible. It's sometimes helpful to take a systematic approach—moving from the upper left to the bottom right of the UI—to make sure you haven't missed anything.

Apps have a first-time experience (no user data) which typically includes messaging prompts, they also have a subsequent-time experience (with user data) which can look and behave differently. You'll want to set up the app so you can experience its full potential. This might mean creating a profile, entering placeholder, or "dummy" content, or inviting friends to connect.

Be conservative when performing destructive actions, they may not be reproducible again later. For example; removing a file, deleting a friend or account, etc.

Step 2: Capture screenshots

Take screenshots while you're moving your way through the app. Be sure to capture everything—especially so called "onboarding experiences" which may not be reproducible later—you never know what you'll need so better to be safe. Not sure if you already captured a screen? Go ahead and capture it again. It's easier to delete a duplicate than it is to go back and reproduce the steps to get to the screen.

Rename the generic screenshots and give them unique names. This takes some time, but it's easier to locate specific screens later, which you'll surely need to do. All the screenshots should be located inside a single folder to keep file paths simple. Sub-folders complicate things needlessly.

Capturing Capture screenshots using Shift + Command + 4 + Space Bar on OS X, or the Home + Sleep/Wake buttons on iOS. There's also a variety of powerful, third-party screen capture apps available. I prefer to exclude drop shadows from desktop windows because they're visually distracting, and increase file size needlessly. A simple Terminal command banishes them from your captures. A web search will point you to detailed instructions.

Naming I prefer not to name my screenshots after the information architecture of the app because if the app is confusing, then your files will be too. Instead I name the screenshots based on what they are.

For example;

– tab.png
– tab_section.png
– tab_section_widget.png

I prefer to name my files in lowercase with underscores between words. Underscores signify a move into a deeper level of the app. You can name your files however you like—Initial cap, InterCap, hyphens, etc.—the important thing is that you're consistent.

Step 3: Layout the screenshots

Using the app should give you a mental model of how it works, which could lead to layout structure. For example, should the screens be organized horizontally?, vertically?, or a combination of both? Should the home screen be located on the left?, right?, or center? Save time by sketching a rough structure before moving to the computer. This is harder than it sounds, but this rough blueprint will save you precious time when you move into detailed digital work.

Grid Some trial and error will be needed to determine your document grid, i.e., how many rows and columns of screenshots can fit into your document. Be mindful to leave room for the containers, screen titles, notes, and the multitude of lines which connect the screen together.

I prefer to use a 12-point grid with snap to grid activated. You can set your document grid through the menu: Illustrator > Preferences > Guides & Grid. Set Gridline every to 72 point, and set Subdivisions to 6. 72/6 = 12. You can also activate snap to grid through the menu: View > Snap to Grid.

All objects (screens, typography, and lines) fall on the grid. Even the type sizes and leading are derived from this measure of 12, for example, text type is 10/12 point. 12-point is flexible enough to work with, while still remaining cohesive. A finer grid doesn't provide enough structure, while a looser grid isn't flexible enough.

Linking Screens should be linked to, and not embedded into your Illustrator file. Linking enables you to swap out screens easily, and produces a significantly smaller file size—which is important when printing or sharing your map. The only downside to linking is it requires managing the folder of linked screenshots, but it shouldn't be a problem if you follow the tips described in step 2.

Scale Screens should never be placed at 100% (especially in the age of high-resolution displays). Screens from mobile devices should be scaled down to simulate physical size (some math will need to be done to determine the correct scale percentage for your specific device). Place and scale your first screen, then use the Move tool to copy it, and simply relink the file in the second instance. The Move tool automates what would be a tedious, inaccurate, and accident prone manual task.

Move Tool Use the Move tool to move and/or duplicate objects. It's more accurate and consistent than dragging objects around the canvas. Select an object, go to the Object menu, then Transform, and select Move (or use the Shift + Command + M keyboard shortcut).

Strokes Sometimes screens may require a 0.5 point stroke outline to separate them from the background. Again this speaks to contrast. For example, if the screenshots are mostly white, and they're placed on a white background, then adding a stroke helps to separate the screens from the background. Strokes can be drawn manually, or added as a graphic style.

Cutoffs and Crops Tall scrolling pages, say a long blog post, can be cut in the middle so they don't disrupt the grid of your map. A variety of visual affordances can be used to signify to the reader that a screen is cut.

Similarly, if only a small section of a screen is changing, you could crop into the specific section to help the reader focus on what's important.

White Space Don't be afraid of white space. Although being conscious of paper and ink consumption is nice, it shouldn't be your main goal; making the map clear and logical should be.

Step 4: Group sets of screens

Containers provide hierarchy, and entry points into the content.

The app surely contains levels of hierarchy. This hierarchy can be visually illustrated to the reader through containers—boxes which hold sets of screens and show the organization structure at a high level.

Color There's no absolute rule for color choice; it depends on the design of the app. The goal is to have enough contrast so the screens stand out from the containers, and the lines stand out from the screens. For example, if the screens are primarily gray, then don't use gray containers.

Containers can be organized into different sets of colors if it's appropriate and adds value. For example, login, signup, and onboarding can use blue containers, while the main content uses green containers, and settings uses gray containers.

I recommend that the largest containers are darker, and the subsequent nested containers get progressively lighter. Light to dark may make sense conceptually as you are adding levels, but having the containers get lighter is better for legibility.

Step 5: Add typography

Each screen should have a title. Sometimes you'll also want to add a note. In addition, each screen (and container) should have a unique number to make it easy for readers to provide very specific feedback.

Numbering System The hierarchy of the interface should be reflected in the both the numbering system and the containers. They should be thought of as connected; if the interface goes down a level, then a new container and a new screen digit should be added. Once you reach the screen level, append a letter. One could imagine a double letter system, but I've never run into the 26-screen-per-container limit of this system.

For example;

– 1	Container
– 1.1	Container
– 1.1.1	Container
– 1.1.1A	Screen
– 1.1.1B	Screen

Step 6: Connect the screens

Connecting screens with lines is the most tedious part of mapping. They're all drawn by hand (no automation yet) so it should be saved for last because you don't want to have to redraw lines because you discovered a new screen or a screen moved. Lines are drawn with the pen tool and aided by the snap-to-grid setting. The result is consistent spacing and placement; a uniform appearance.

Notes frequently accompany lines. I prefer to make the notes typographically different from the main body copy and color-code them so they're the same color as the lines so readers make a visual connection.

Line Types There's a variety of line types. The primary line type is solid with a dot (represents a click or tap), and an arrow (represents flow direction). If there's no click or tap, for example, an amount of time passes, then there's no dot. Or perhaps you want to callout a detail, then there's no dot or arrow. If there's an alternate path or want to de-emphasize a line, it can be dashed. Alternately, you can emphasize a line by adding a thicker line beneath it.

Sometimes lines cross each other and makes reading direction difficult to follow. Making a line jump over another (like a bridge overpass) keeps the connections clear. I prefer to keep all the jumps in the same orientation; for example, left-to-right, or up-and-down, but never mix both.

Sometimes a path can split. This is indicated by a decision diamond. The path enters one point, a question is asked, and the path continues out of the Yes and No points. Decision diamonds are flexible; for example, the question can be on the top, bottom, left, or right, and the Y and N can be on whichever axis is appropriate.

Use the Stroke palette to control line weight, line styles, arrowheads, and scale of arrowheads.

Rounded Corners Avoid adding rounded corners to lines. I've found that the effort is not worth whatever incremental decorative gains you might find. Let me explain my hesitance.

Imagine a series of lines which all make a 90 degree turn around a corner. If you were to apply a consistent corner radius to those lines, you would see uneven—and unattractive—negative spaces between the lines. To achieve consistent spacing—and a harmonious appearance—the line to the outside must increase it's corner radius (think Nascar race track).

Even if you took the time to mathematically adjust the corner radii, what was the outer most line can—and often does—become the inner most line (think S-shape), making your previous effort null and void. Do yourself a favor and avoid this issue altogether and stick with sharp corners.

Conclusion

I've introduced the idea of mapping a software application, explained its value, and outlined my process for doing it quickly and efficiently.

This guide is by no means the only way to do things, but it presents what I've learned as best practice. I like to think of it as a framework from which to work. New problems will arise, and new solutions will be needed.

I hope you find this guide useful, and I welcome your feedback.

Appendix: Thoughts on managing complexity

Multiple Passes You're juggling a lot of elements and it's easy to get distracted. I find it's helpful to take things one at a time and make a series of passes, focusing on a different task with each pass. For example, on this pass, I'll add the screen titles. On the next pass, I'll add the numbering system. On another pass, I'll add notes. I've found that this helps reduce mistakes and keep quality high. The map becomes more refined with each pass.

Layers Adding layers to your Illustrator file is another way to enhance working in multiple passes. What better way to focus on a single task, than to make the others disappear?

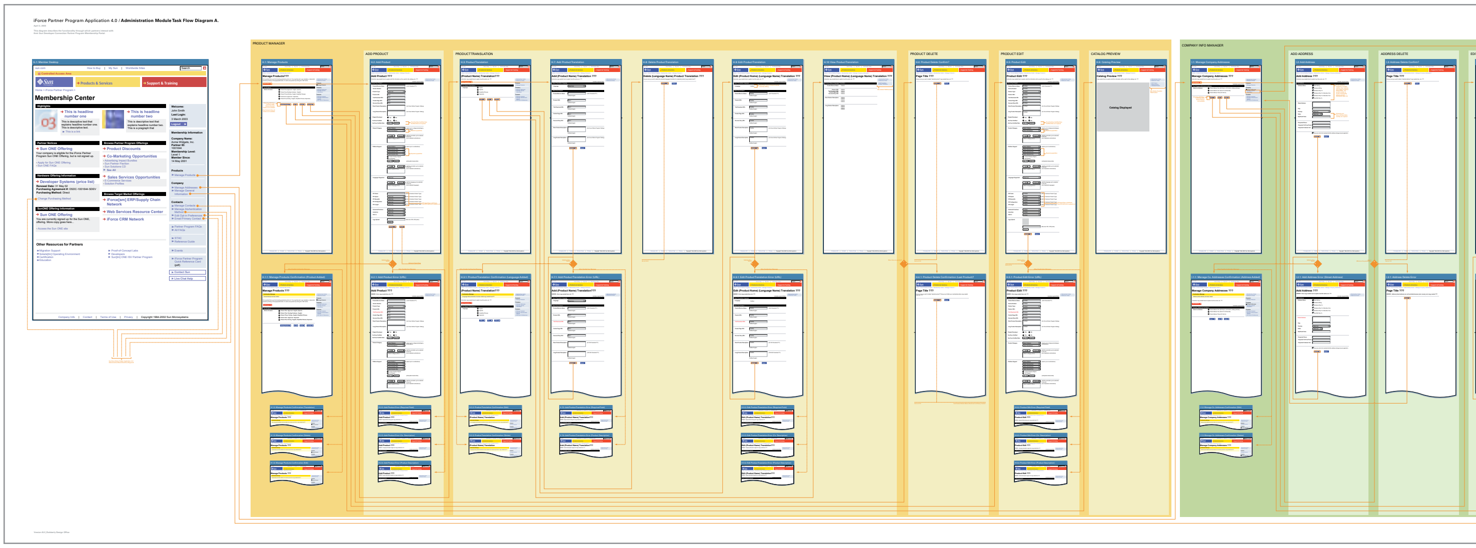
Here's how I set up my layers (you'll notice the relationship to the steps outlined in this document).

- Typography
- Lines
- Screens
- Containers

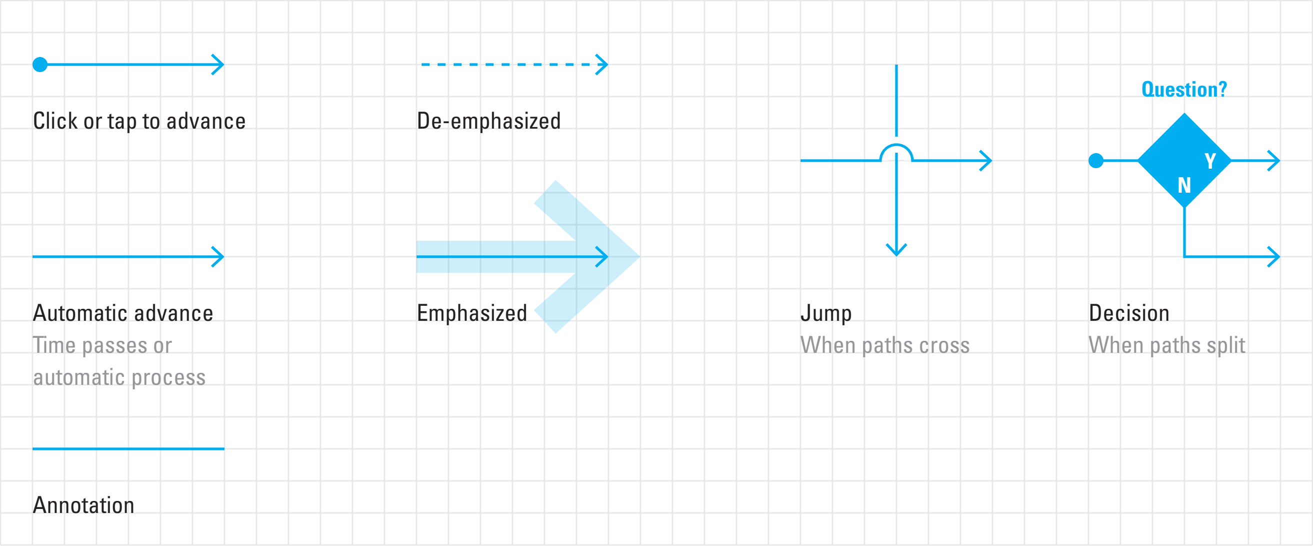
Here's how I use the layers; when I'm laying out the screens, I turn off the typography, lines, and containers layers. When I'm adding the numbering system, I turn off the lines, and containers, but keep the screens layer visible.

I set up my Illustrator files with minimal layers, and like I previously mentioned in step 2, stick to one level of hierarchy for the sake of simplicity. If you add sub-layers, I've found that you spend more time fiddling with layer management (which is invisible in the final document), than actually working with the content (which is visible in the final document).

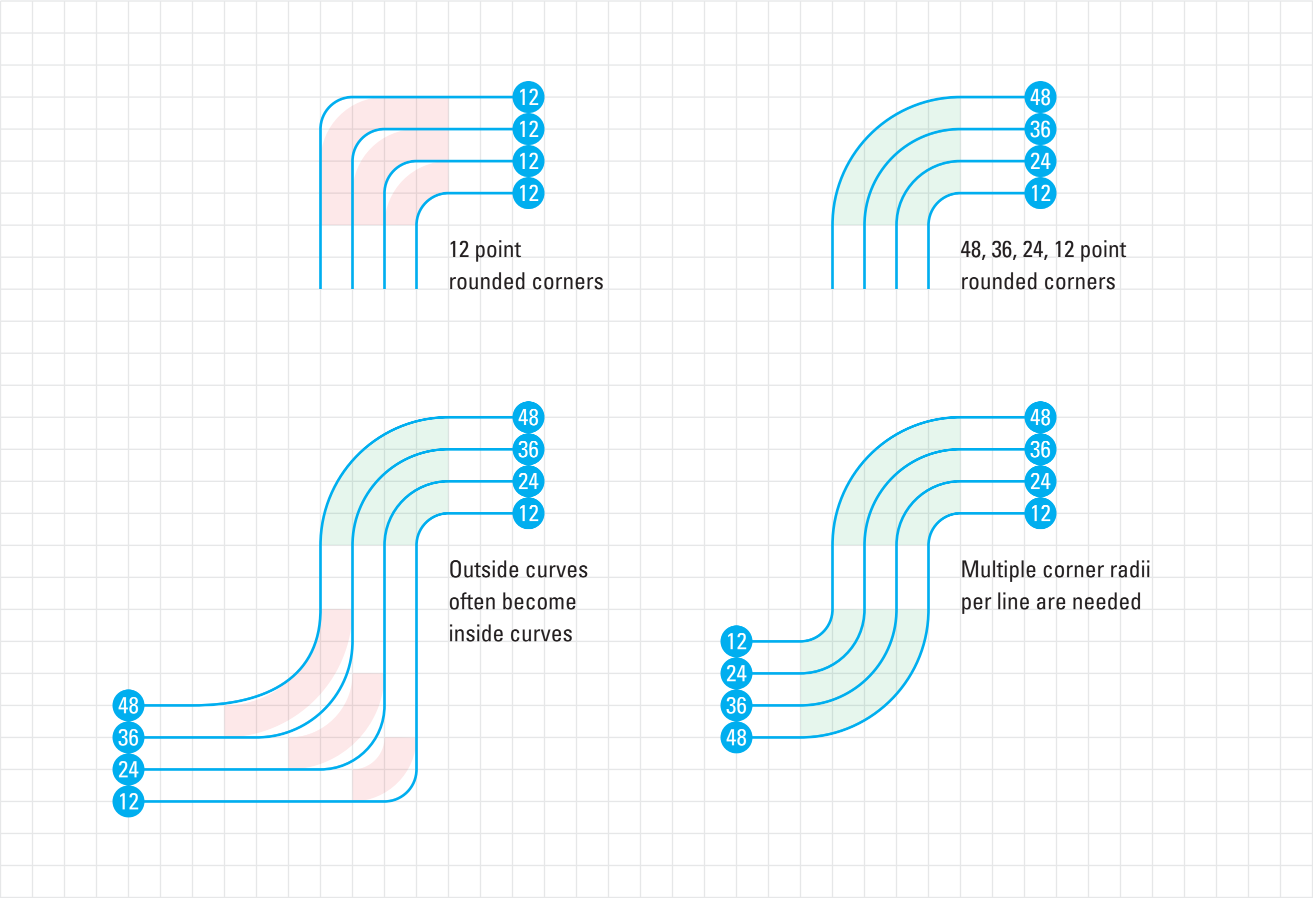
Many techniques, one map
You can see a variety of techniques being used in this example map. Multiple screen sizes show hierarchy. Long scrolling pages are cutoff to optimize space. Screens are cropped to help the reader focus. Different sections have different container colors.



Line types
There's a variety of line types available to help you solve the most common problems.



Rounded corners
It takes a lot of manual work to get rounded corners right.
Save yourself a lot of time and stick with sharp corners.


12
12
12
12
12 point rounded corners
48
36
24
12
48, 36, 24, 12 point rounded corners
48
36
24
12
Outside curves often become inside curves
48
36
24
12
48
36
24
12
Multiple corner radii per line are needed
12
24
36
48