Foundations

Guide 0.8

Spring 2015

Precepts of this guide

It’s your call

This information is presented as mathematical, verbal, and visual statements.

This information functions as a means of informal design guidance, subject to norms and processes, and is maintained through the intricate network of voluntary participation by development, including programming, testing, and project management, and enforced by the people themselves.

This work belongs objectively to those that use it.

Context specific

The guidelines presented here will not apply to every situation encountered or required by users.

However, the general concepts and principles of this guide will help solve the visual communication problems specific to the context under development.

There are fundamental problems, and fundamental solutions to those problems, because the human aspects of the context related to sensory input do not change.

Realizing the basis of the problem and finding the problem solved elsewhere in another basis will uncover the solution; through direct indication or through inspiration.

Ultimately, the user’s situational workflow determines the interface.

Vision

Purposes

Make development easier and quicker

Promote consistency

Provide inspiration

Content

Some sections are not part of the preview.
All sections will likely evolve with time.

Principles

Layout — examples | canvases | lists | grids | detail | tabs

Control — checkboxes | switches | radios | sliders | text input

Text — voice | messaging | capitalization

Dialog — menus | forms | notifications | tooltips

Artwork — buttons | icons

Color

Principles of the interface

A living definition for an interface

An interface is a constructor that maps a set of inherited functions onto any set of given data
to allow for impulse, action, and potential cognition.

Content first

Our work is user-centric, context-driven, and task-focused.

Balance the presentation of information with the ability to act on it; emphasize content over the supporting structure.

Images read faster than words when the image is simple; if the concept is complex, consider using text instead.

Ease and visibility of action on content

Essential, frequent, and typical actions should be seen quickly and be easy to apply.

Show as much information as cleanly as possible.

Group like actions, events, and notifications with each other and with their relevant information for fast processing.

Mobile in mind

Some things will be done on the desktop, but try to keep the mobile format at the top of mind.
The mouse, keyboard, touch, and voice are all primary inputs.

User interactions for different form factors and usage of space are key considerations.

Principles of software design

Think workflow

The user workflow can be thought of as a series of steps; breaking these steps down and focusing on what is most important at each point will help create the best design. Consider a task centric UI flow that promotes learning:

Presentation of information, user action on information, feedback event

Visual hierarchy

Arrangement of information so that it communicates a clear message and facilitates ease of action on that information is paramount to successful design.
Success can be gauged by asking: “What do I see first, is that what I want the user to see first?” and “How can I group the given information to help the user make quick sense of it?” and “Is it easy to act on the given information?” and “Does the presentation match the user’s expected workflow?”

Three more things

Three things to consider when questioning the efficacy of a design:

Contrast —— similar objects recede, while dissimilar objects will stand out

Composition —— layout can point the user toward areas of importance

Conservation of energy —— reduce the cognitive load of the user and strive for simplicity

Principles of software design, continued

Employ imagery, when possible and appropriate

Images are read and understood faster than text in many cases and can help during the localization process to get the right message across. Images can also add to a positive user experience.

Cautions with imagery

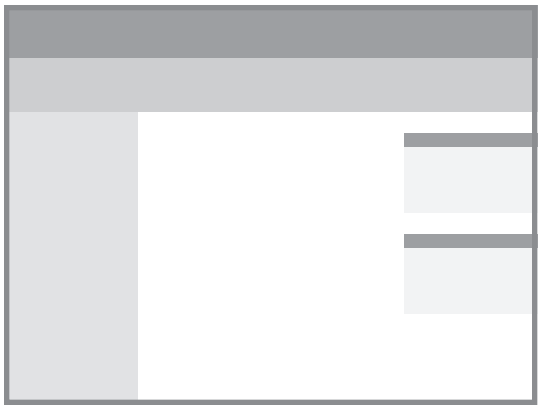
Not all images are interpreted the same by different locales. If an image is not quite common, of a hand gesturing with two fingers for example, be sure to test the image with users from the different locales.

Imagery can also clutter the interface quickly; use it keeping the balance of a clean interface and a quickly read one in mind.

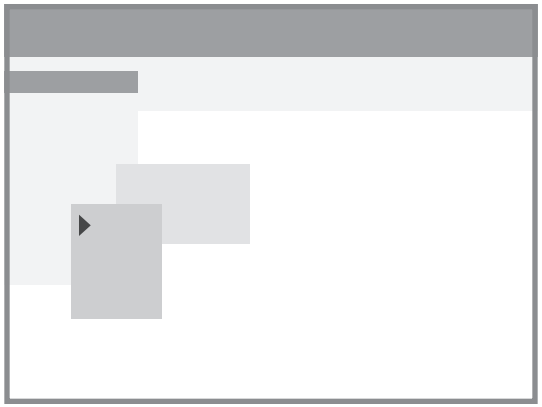
Use imagery to educate

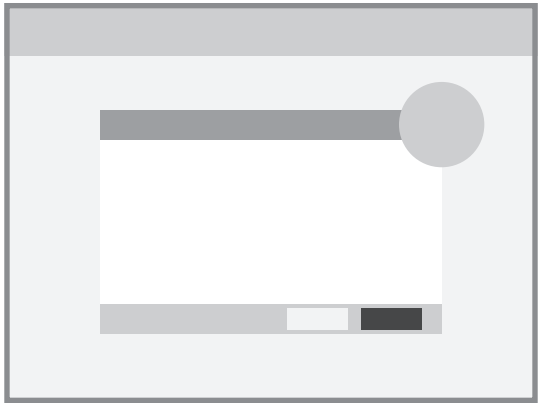
If the same images are being employed throughout the interface, their repetition can be utilized as a method to promote quick recognition and help speed up a user's workflow. Images will also help describe difficult concepts.

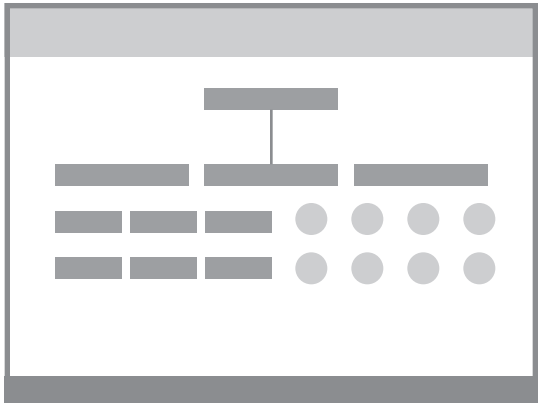
Layout: Examples, simple



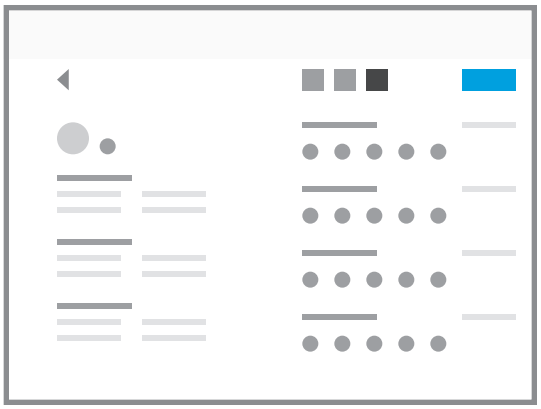
Layout: Examples, complex







Layout: Examples, platform



Layout: Critical grid elements

Units of 4

Think in increments of 4dp (density-independent pixels) with gaps between panels of at least 8dp.

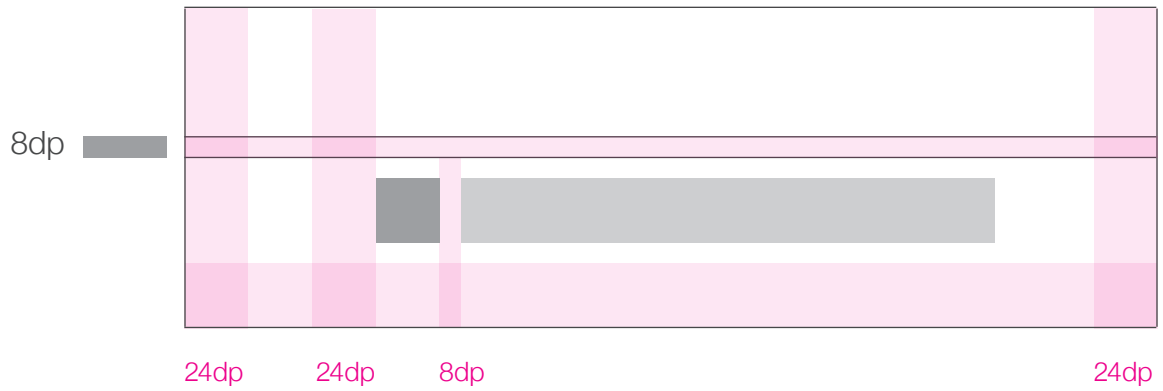
Importance of margins and whitespace

Consistent margins will help to establish consistency throughout the product. Breaking up the grid with 8dp or 16dp ‘visual breathing gaps,’ or whitespace, is a method of building focus for the user on what is important.

The 24dp left and right margins equate to about 1/4,” which is comfortable spacing in the desktop, tablet, and phone contexts. Having the mobile context in mind, consider finger size, and holding and usage patterns.

Size and action potential

Equivalently spaced areas inherit and imply the importance of information; and the ability to act on the information. Consider an icon, which is small in size and highly actionable, and a title bar, which is large in size and not typically actionable though it can contain actions. The narrowing of focus raises the potential for action.



Layout: Icon elements

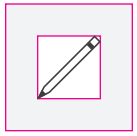
Touchable, satisfying

There is a delight associated with the successful press of a button and with a perceived success based on the size of the touchable target.

The recommendation here is to match the target size to the size of a fingertip. In pixel dimensions, this is about 48 pixels squared. With this target size, an icon within this gap should be between 20-24 pixels squared.

An adaptive need; keeping inflation down

Desktop applications may appear inflated with large icon target areas and margins. If this aesthetic is not the desired one, try making the entire target area 24 pixels squared and allow the icon to consume this entire space. By reducing the airy gaps, the design will appear tighter and more information can be packed into the interface envelope.



48dp^2 target area

24dp^2 icons

Layout: Canvases

The information landscape

It is useful to break the user interface into sections such as the canvas to help architect the layout of information. This separation is as useful to organize what should appear as what should not appear- or could appear later.

Considering the canvas as the container for dialogs, warnings, notifications, and potentially the main information, it will need to support a variety of controls, containers, and further break downs of the interface.

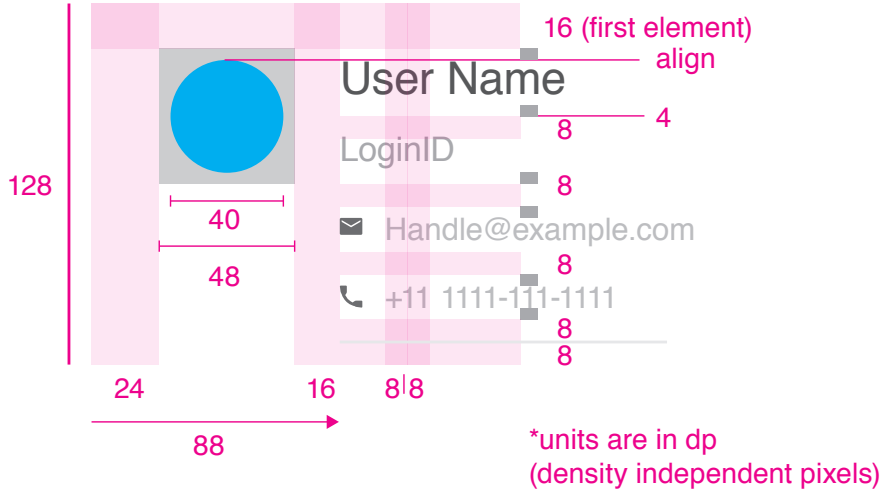
In this way, it is crucial to address the scalability of the canvas container and its elements, especially in the contexts of different form factors such as a phone, tablet, TV, computer monitor, or wearable screen.

Formulas

Standardizing on the layout of the canvas and following through on this formula will nurture consistency and unification of the product. It is important to determine the exact locations of action and navigation buttons and how dialogs and notifications will appear early on during creation of a product.

Layout: Lists

Details



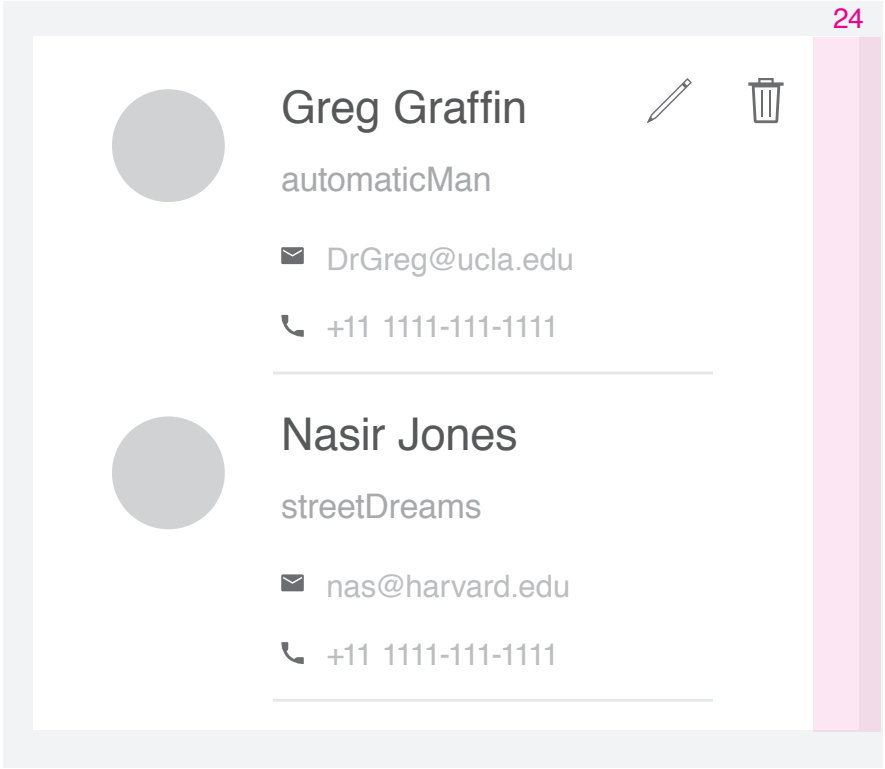
User Name: 16dp

LoginID: 12dp

Handle@example.com: 11dp

+11 1111-111-1111: 11dp

Context



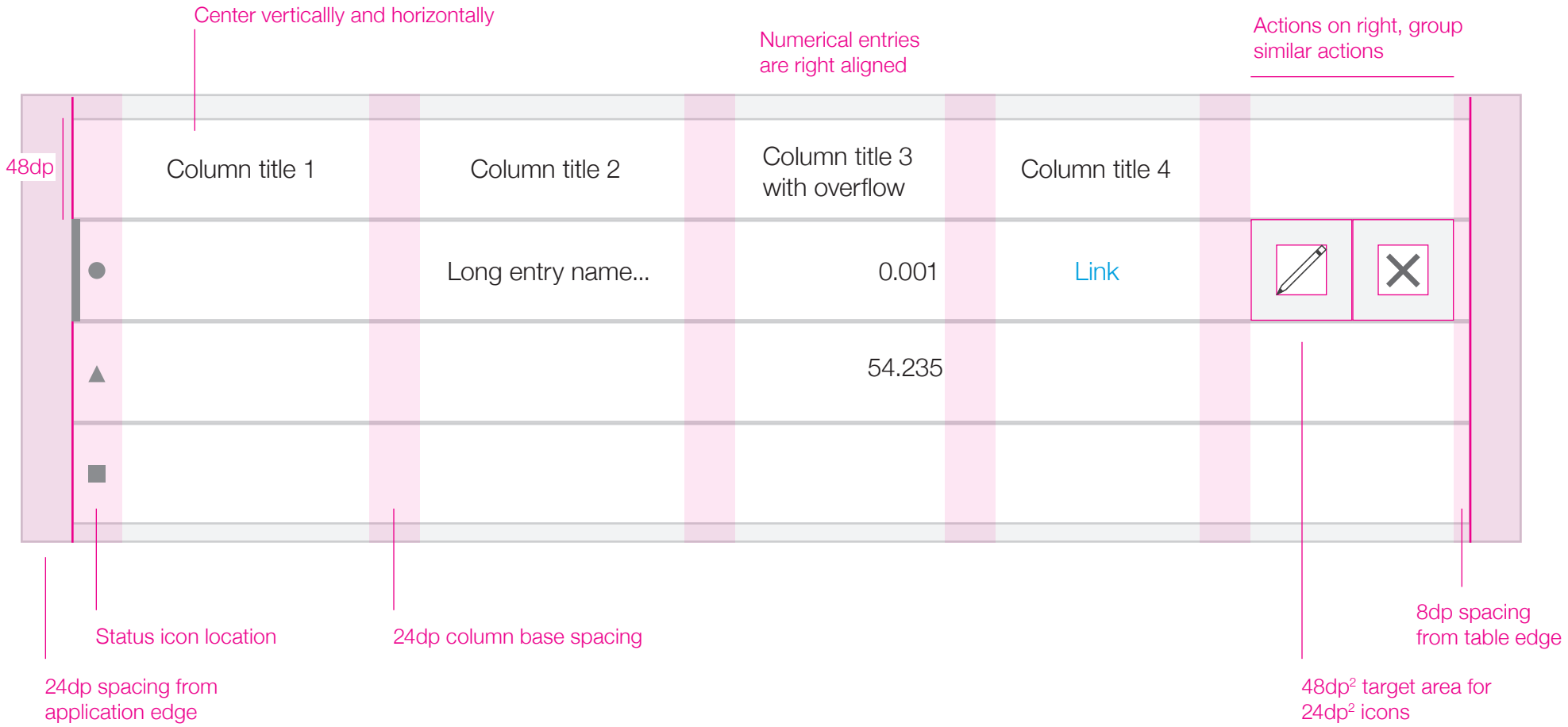
Notes:

Multiple sub titles (e.g. email handle) and meta details (e.g. phone number) can be stacked or arranged in an array horizontally. For the latter case, use an equidistant spacing of at least 24dp between columns. Utilize icons to separate textual information across columns for easy reading at a glance.

Layout: Tabular grid

Arrays of data

Employ tables with care since they can be information-heavy and can limit action on information. Avoid visual clutter in the table by leaving out unnecessary visual elements.



Layout: Tabs

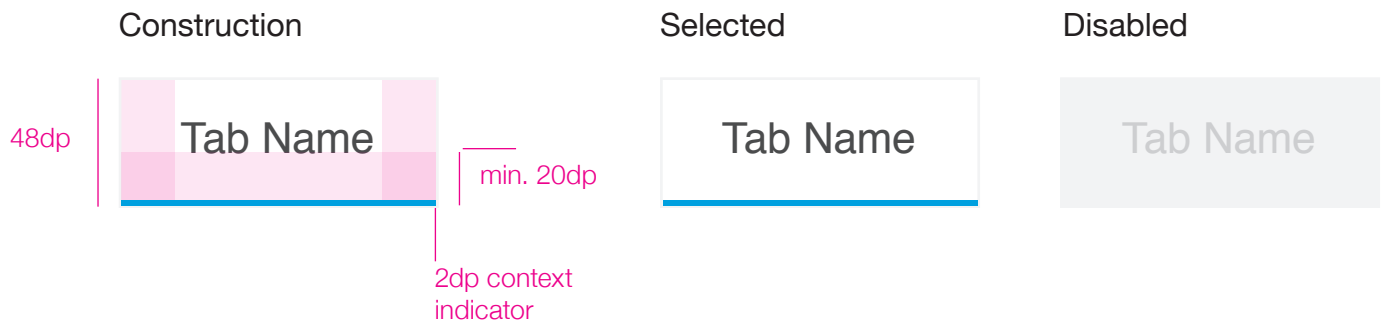
Shallow viewers, equal peers

Use tabs when you expect users to frequently switch views within the same context.
Tab ordering can help establish workflow.

The content that tabs reveal should be limited to a top level; let the user uncover only shallow hierarchies of information, rather than deep ends.

When using tabs, ensure that users are aware of what alternate tabs display by keeping the context confined.

Using tabs in a focused context where a small number of tab peers provide equally important views is a good practice for usability.



Text: Voice

Be human

Remember the audience is human; speak to them that way.

Be as clear as possible

Be direct with software messaging; users will appreciate it. The user is interested in doing something with the information we provide- place important information first and emphasize the ability to act on it.

Don’t blame

In the case of an error message, provide the user with the information needed to fix the problem; use other visual clues to indicate there was a problem.

Be encouraging

Provide feedback on actions in the right proportions. Button actions require less visual feedback than large software changes. At times, one form of feedback will follow another- for example, a button tap followed by a ‘Validating...’ message.

Text: Messaging

Avoid computer-speak

Don’t show messages, titles, or options in code format, camel case, or Pascal case (e.g. snmpBufferCollector), unless the user has specifically named objects that way.
Within the GUI, utilize the text areas for common language.
Choose impactful, meaningful, and context-appropriate, word combinations in an API context that promote memorization for expediency and ease of use. Try to make your strings guessable.

Think locale

Realize that messages will likely be translated.
Use simple words and sentences to increase the likelihood of accurate localization.

Create a formula

When writing error messages, try to construct a repeatable formula for your presentation thinking in terms of subject, condition, relationships, resolution, and scalability of the message.
This method will drive consistency and ease of use across the product.

Error message example

Production_Server_01 has lost communication with TOR_switchA.

Resolution: Check the physical connections at TOR_switchA.

Text: Capitalization

Follow the (current) rules

Employ existing guidelines on capitalization for the appropriate locale.

Capitalize proper names, nouns, places, dates, holidays, etc.

Be consistent

Above all else, be consistent with capitalization; the shapes of the letters are as important as any other shapes on the screen.

Choose a format early on for table headers, dialog titles, input labels, attributes, and other items, and stick to it across the entire product suite.

Document the choices and make them public to drive consistency.

Dialog: Tooltips

More help than harm

Tooltips are useful forms of help and necessary at times, and they should be used as a last resort to solving visual communication problems. Defaulting to the use of tooltips will slow down the user's flow and create a massive learning hurdle from the very beginning. The abundant existence of tips can become a user's crutch and they will feel less empowered to discover and learn the product on their own.

Capitalize the first word and any proper nouns within the tip.

No period is necessary unless the tip is a complete sentence.

Verification and validation tooltips have periods since they are constructed as sentences.

General tooltips are not sentences and should not have periods.

Artwork: Buttons

Order and location matter for consistency and workflow

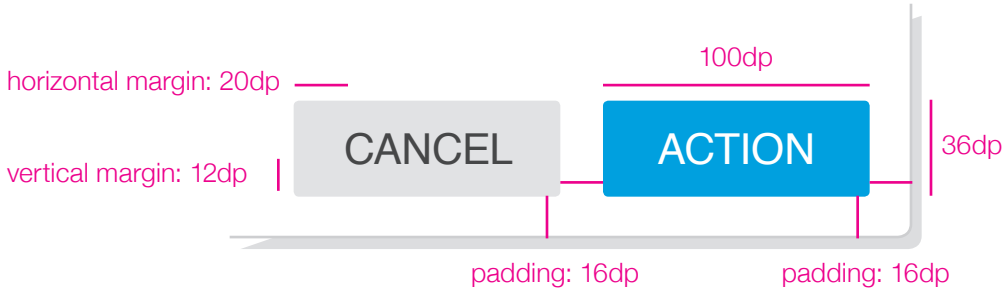
Here, the most impactful buttons are on the right and cancellation-like buttons are on the left- this allows the user to continue along a left-to-right workflow. This button ordering can also be thought of as backward (cancel) and forward (action); in this way, ‘Cancel’ makes the most sense on the left.

Place buttons most near the information their actions refer to- think in terms of conservation of energy for the user.

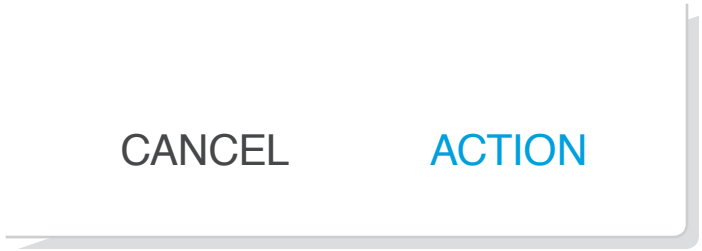
Consistency of button placement is paramount to prevent accidental incurrence and impedance of flow.

Example: Dialog actions

Floating buttons are easy to find in a dialog with a lot of text and add dimensionality to the interface.



Flat buttons make dialogs cleaner by removing dimensionality, and by minimizing color and shapes.



Artwork: Buttons

Visual overload

Try not to overload a button with more than what it already is and does.

A colored rectangle, colored text, or actionable icon are examples of buttons on their own. It is not necessary to combine icons, colored shapes, and text to communicate the message that ‘this can be acted on to do something.’

Weighting

Consider the weight of the button object within the ambient context and ask ‘How important is the presentation of this action compared with the rest of the screen?’ or ‘How destructive, potentially, is the action caused by the button?’

Placement establishes flow

Buttons that can be grouped by similar function, like editing actions or form submissions and cancellations, should be placed in the same place throughout the interface.

This intentional formatting and placement develops a workflow for the user that is either learned or intuitively understood. This consistency will drive a feeling of unification for the project.

The conscious effort devoted to establishing place for actions will also precipitate the answer to what the most important information is and how it should be presented to the user. This is simply a return to the origin of the need for the project in the first place: The user wants to do something, what is displayed that allows for that action.

Artwork: Icons

Shape matters

When creating icons for a product or suite of products, consider the big picture and how all of your icons can be viewed as a cohesive set.

Using the same sizing grid and simple rules for each icon will create an inherent consistency from screen to screen. This patterning and conscious intention will be apparent to the user.

Examples are the use of only ‘2-D’ shapes, not having rounded corners on stroked shapes, or limited color usage.

Size your icons according to the screen context that is most appropriate.

Examples


Directory


Dashboard


Shared folder


Hypervisor


Memory


Timer


Magic wand


Copy to clipboard


Remove

Color

Use with care, and intention

Color is an effective communication tool, but all designs should communicate sufficiently in greyscale.
In this way, utilize color as an accent to communicate something quickly and strongly with intention.

Many colors inherit connotations that are difficult to decouple from the meanings they provoke- the color red is an example. Colors also take time to mentally process, so keeping them to a minimum will help maximize user focus.

Visual impairments, like color blindness, should be considered at all times during production.

Suggestions

70% Grey
#bbbdc0

Icons

85% Grey
#4c4c4e

Text

15% Grey
#dbdcde

Disabled text, separators

#00a0df

Links, buttons, graphs

#55ba5a
#ff5450

#ffe019
#ffa310

#58c2f7
#3772b9

Status color indicators- individual colors should stand out in a sea of their nearest color neighbor, even at small resolutions; and polar colors should appear to fluoresce when juxtaposed given the polarity of status.

guide