

RTI callouts

Best practices guide

Contents

Goals & Flow	3
Layout Guideline	6
Fitt's Law.....	7
Weber's law.....	7
Gestalt Principles of Grouping	9
Proximity	9
Similarity.....	10
Continuity	10
Law of common region (Palmer).....	11
Good Form	11
Consistency	12
Miller's Law of Short Term Memory Load.....	13
Navigation Guidelines	22
Tabs.....	22
Horizontal Menus.....	22
Trees	22
Drill Throughs?	23
Direct Links.....	23
Breadcrumb Trails.....	24
Usability testing	25
Controls	27
Check Boxes.....	27
Radio Buttons.....	28
List Boxes	29
Drop-Down List Boxes.....	30
Drop-Down Combo List Boxes (i.e. Combo Boxes).....	31
Command Buttons	32
Date Pickers	33
Text Boxes.....	34
Spinners.....	34
References.....	35
UX Do's and Don'ts – quick tips	36

Good UX.....	36
General.....	36
Text.....	36
Graphics.....	37

Goals & Flow

Flow, as a mental state, was first proposed by psychology professor [Mihaly Csikszentmihalyi](#).

It's a state of being that makes an experience genuinely satisfying, in which emotions are aligned with the task at hand. Everybody has experienced it. Most people refer to it as being "in the zone" or "in the groove". We want our users to experience it as much as possible

10 Components of Flow

Csikszentmihályi identified 10 experiences that go with the state of being in flow:

1. Having a clear understanding of what you want to achieve
2. Being able to concentrate for a sustained period of time
3. Losing the feeling of consciousness of one's self
4. Finding that time passes quickly
5. Getting direct and immediate feedback
6. Experiencing a balance between your ability levels, and the challenge
7. Having a sense of personal control over the situation
8. Feeling that the activity is intrinsically rewarding
9. Lacking awareness of bodily needs
10. Being completely absorbed in the activity itself

Not all of these factors and experiences necessarily have to be in place for flow to happen, but you're likely to experience many of them when flow occurs.

3 Conditions

Csikszentmihályi also identified 3 things that must be present if you want to enter a state of flow:

1. **Goals** – Goals add motivation and structure to what you're doing. Whether you're learning a new piece of music or creating a presentation, you must be working towards a goal to experience flow.
2. **Balance** – There must be a good balance between your perceived skill and the perceived challenge of the task. If one of these weighs more heavily than the other, flow probably won't occur.
3. **Feedback** – You must have clear, immediate feedback, so that you can make changes and improve your performance. This can be feedback from other people, or the awareness that you're making progress with the task.

Friction

When the smooth path is interrupted, or something doesn't seem to fit, users notice and the flow is broken, which means that the experience is also momentarily broken. These small episodes of friction are cumulative. Unfortunately, the breaks in flow weigh more heavily on the experience than the positive, frictionless moments.

Clutter, animation, and surprises may interrupt and be disruptive. Online, people don't like surprises (especially the kind where they go "now what?", "how do I..." or "what's that?"). Take out or improve that might cause friction. Less is more: remove visual and navigational noise that might seem like clutter to users.

The UX Designer

The three conditions are good news for the UX designer, because we heavily influence all of them in the following ways:

1. **Goals** – Either the user's supervisor or the user themselves generates the high level goals. The product interface has to acknowledge those goals and tasks required to achieve those goals. There has to be a clear on-ramp leading to the completion of those goals and tasks.
2. **Balance** – The user experience can help achieve a balance between user skill and the challenge of the task by including easily accessible user guidance and prompts. Tasks can be broken down and cast as easy to follow wizards. Similarly, wizards and help can be tucked out of sight when it is no longer necessary and becomes a point of friction. Furthermore, more advanced and efficient paths to achieving goals should be discoverable, and elements of Gamification can help increase the challenge of performing tasks that have become routine for the experienced user.
3. **Feedback** – Ensuring the system is responsive to the user's input, alerting the user to changes in the state of the system that they or other actors have caused... these are straightforward design choices for us to make.

Many of the 10 components of flow are also within our influence:

1. **Having a clear understanding of what you want to achieve** – Addressed by having a clear understanding of our users' goals, and providing clear navigation to the tasks and data required to complete those goals
2. **Being able to concentrate for a sustained period of time** – Addressed by elimination of friction causing distractions on the screen
3. Losing the feeling of consciousness of one's self
4. **Finding that time passes quickly** – Addressed by having informative and distracting progress indicators for longer tasks
5. **Getting direct and immediate feedback** – Addressed by building in plenty of immediate feedback
6. **Experiencing a balance between your ability levels, and the challenge** – Addressed by balancing the complexity of the visual design with the complexity of the task. Simple tasks should appear simple, and complex tasks may appear more challenging. Goal directed usage patterns, as all of ours will be, should be calming, not arousing.
7. **Having a sense of personal control over the situation** – Addressed by building clear choices with indications of consequences into the tasks
8. **Feeling that the activity is intrinsically rewarding** – Addressed by showing progress toward and acknowledgement of the completion of goals
9. Lacking awareness of bodily needs
10. **Being completely absorbed in the activity itself** – Also addressed by the elimination of friction

References

Much of this has been liberally appropriated from the following sources:

[http://en.wikipedia.org/wiki/Flow_\(psychology\)](http://en.wikipedia.org/wiki/Flow_(psychology))

http://en.wikipedia.org/wiki/Mihaly_Csikszentmihalyi

http://en.wikipedia.org/wiki/Yerkes-Dodson_law

<http://conversionxl.com/how-to-design-user-flow/>

<http://www.mindtools.com/pages/article/newlmd/112.html>

Layout Guideline

Designing the user experience is both art and science. Over decades of research, several famous laws and principles have emerged. Here are the important takeaways.

1. Check box and radio button labels must be included as part of the click target, making that target as big as possible
2. Check box and radio button labels are always to the right of the box or radio button.
 - a. A notable exception is when they appear as part of survey, in which case the label is on the left, and the series of rating radio buttons are on the right
3. Actions associated with checked items in a grid should appear close to the column of check boxes. This is usually on the “leading edge” of the grid (e.g. the left side for European languages)
4. The visual design at rest should be calm and quiet. Subtle differences in state should be reflected with subtle visual cues. More prominent changes in state may use more prominent visual cues.
 - a. Suggested examples:
 - i. Freshly rendered (so calm and low contrast)
 - ii. The hover state for all widgets with hover states
 - iii. The selected state
 - iv. Showing an error message and the highlighting the offending control
 - v. Showing a completion message
 - vi. Showing a system notification
5. Labels on widgets should either be top or right aligned (again, for European languages), keeping them as close to the widget as possible.
6. Information elements that have a strong affinity with each other around an appropriate theme should be grouped closely together, like the different text entry fields that comprise an “address” (like Street, City, and Country)
7. Be as consistent as possible. If you have a choice to use an existing template, or copy an existing screen, then do it. If you have to come up with something new, then make it indistinguishable from what already exists.
 - a. Your touchstone should be the design pattern library available on the wiki.
 - b. If you do have to invent something not covered by an existing screen or example, then make sure you usability test it.
8. Grids should use alternating row colors to drive home the atomic nature of the information therein contained.
9. Minimize the need for users to remember things.
 - a. If the user needs to know several items of information in order to make a decision or perform an action, display all of that necessary information simultaneously with the choices for action.
 - b. Error messages and other alerts should restate the context of the error or situation, that is, what is wrong, what has happened, etc.
 - c. Conformation messages should restate what action was requested and completed
10. Alignment rules below assume a left-to-right language.
 - a. Labels relative to their control
 - i. Labels should be above the control if you have sufficient vertical space
 - ii. Labels should be right aligned to the left of the control if vertical space is at a premium
 - b. Vertical alignment of text, graphics, and controls comprising a line/row should be by the baseline of the first line of text (if there is wrapping)
 - c. Dialog buttons at the bottom of a dialog should right aligned

- d. Text in grids
 - i. Heading alignment always matches the alignment of the column values.
 - ii. Text columns should be left aligned. Text includes alphanumeric strings intended not to be interpreted as quantities, such as a serial number.
 - iii. Numeric columns should be right aligned
- e. Page, dialog, and section titles should all be left aligned

References

Much of this has been liberally appropriated from the following sources:

<http://www.measuringusability.com/blog/hci-laws.php>

<http://nanlee.wordpress.com/category/theories-in-hci/>

http://en.wikipedia.org/wiki/Gestalt_psychology

http://en.wikipedia.org/wiki/Principles_of_grouping

Fitt's Law

Paul Fitts described a way to mathematically predict how long it will take to "acquire" a target based on its distance and size. This law states that the time required to rapidly move to a target area is negatively proportional to the distance to the target and positively proportional to the size of the target. Thus, the closer and bigger it is, the faster it is to click on.

Implications for Designers

When applied to interface design, this means that click targets should always be as big as possible, and that action targets should be as close as possible to the means of selecting operands of the action.

Examples

1. Check box and radio button labels must be included as part of the click target, making them as big as possible
2. Actions associated with checked items in a grid should appear close to the column of check boxes. This is usually on the "leading edge" of the grid (e.g. the left side for European languages)

Weber's law

Weber's law states that the ratio of the increment threshold to the background intensity is a constant. So when you are in a noisy environment you must shout to be heard while a whisper works in a quiet room. And when you measure increment thresholds on various intensity backgrounds, the thresholds increase in proportion to the background.

Implications for Designers

Weber's Law means that in order for a difference to be perceived, the difference must be relative to the current magnitude. This can be used for various sensory modalities in GUIs such as brightness, loudness, line length, visual weight of fonts in

typography, color matching etc. If we think of the visual design of our user interface as that noisy room, we must first strive to keep that room as quiet as possible. Then, when we have to draw attention to something, or even several things at once, we do not need to shout as loudly.

Examples

Leonard	Agent		
Bill	Supervisor		
	Supervisor		
Kate	Administrator		
Ben	Agent		
	Agent		
Tom	Agent		
Kamron	Agent		
	Supervisor		

Text	Text	Text
Text	Text	Text
Text	Text	Text
Text	Text	Text
Text	Text	Text
Text	Text	Text
Text	Text	Text

Lorem ipsum dolor sit amet, consectetur adipiscing elit. ✕

Lorem ipsum dolor sit amet, consectetur adipiscing elit. ✕

Lorem ipsum dolor sit amet, consectetur adipiscing elit. ✕

First Name This field is required.

Last Name This field is required.

Selected

hover

Regular

Disable

Payment Summary
✕

Say: “ Thank you for your payment of 143.21\$. Your reference number is 568TY123. ”

Summary was read to the customer

Gestalt Principles of Grouping

Gestalt principles of grouping address visual perception. They maintain that the human eye sees objects in their entirety before perceiving their individual parts, suggesting the whole is other than the sum of its parts. The principles are:

- Proximity
- Similarity
- Closure
- Continuity
- Common Fate
- Good Form

Proximity

The principle of proximity states that, all else being equal, perception tends to group items that are close together as part of the same object, and items that are far apart as two separate objects.

Implications for Designers

Straightforwardly, things that are closely associated with each other should appear physically close together on the screen.

Examples

1. Labels on widgets should either be top or right aligned (again, for European languages), keeping them as close to the widget as possible.
2. Information elements that have a strong affinity with each other around an appropriate theme should be grouped closely together, like the different text entry fields that comprise an "address" (like Street, City, and Country)

User Name	<input type="text"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
E-mail	<input type="text"/>
Phone	<input type="text"/>



User Name	<input type="text"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
E-mail	<input type="text"/>
Phone	<input type="text"/>



Similarity

The principle of similarity states that, all else being equal, perception lends itself to seeing items that physically resemble each other as part of the same object, and items that are different as part of a different object.

Implications for Designers

When it is important to create the impression of a single unit of elements, or to make elements pop out and become more “discoverable”, we can style the elements exactly the same to achieve this.

Example

Notice that the grid is much distinct and easier with banded rows, but the similar icons makes it hard to see the difference between “Report Name” and “Workspace Name”.

	Report Name					Smith
	Report Name	12-Dec-2012; 14:35	John Smith	John Smith		
	Report Name	12-Dec-2012; 14:35	John Smith	John Smith		
	Workspace Name	12-Dec-2012; 14:35	John Smith	John Smith		
	Workspace Name	12-Dec-2012; 14:35	John Smith	John Smith		
	Workspace Name	12-Dec-2012; 14:35	John Smith	John Smith		

Continuity

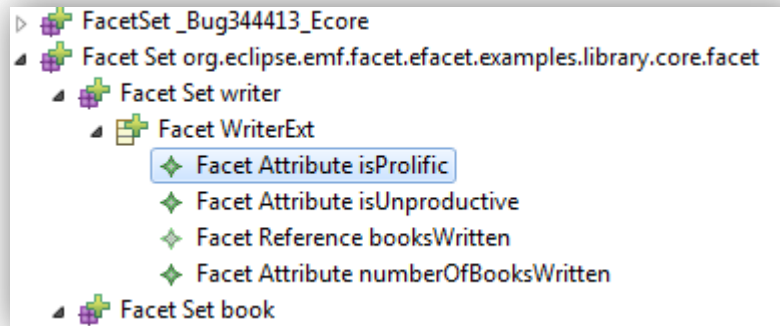
When there is an intersection between two or more objects, people tend to perceive each object as a single uninterrupted object. This allows differentiation of stimuli even when they come in visual overlap. We have a tendency to group and organize lines or curves that follow an established direction over those defined by sharp and abrupt changes in direction.

Implications for Designers

This allows us to be more minimalist with our visual design. For example we need not show dashed lines or the like to indicate the continuation of the edge of a shape when another shape overlaps it.

Example

We can easily recognize that the following items are a hierarchy, since they create an imaginary descending graph. By moving the objects we can visually show which items are on the same level etc.

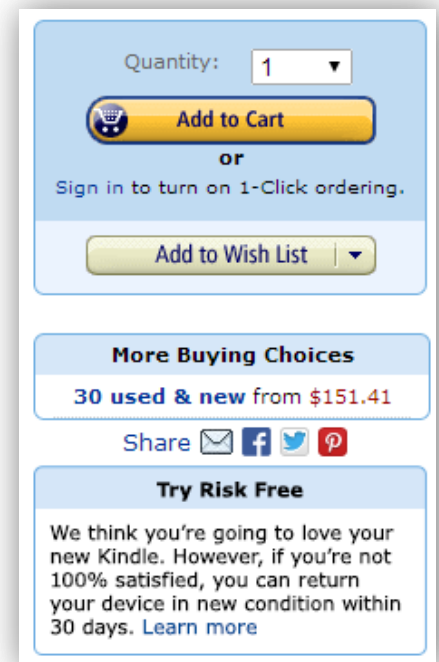


Law of common region (Palmer)

Visual items situated together in demarcated (bordered) regions are perceived as belonging together.

Notice how all the purchase related items are in the same logical and graphical area, the social network items in a distinct row etc.

This makes each component stand out and makes the application easier to use and also allows developers to decide which part are more important and to highlight them.



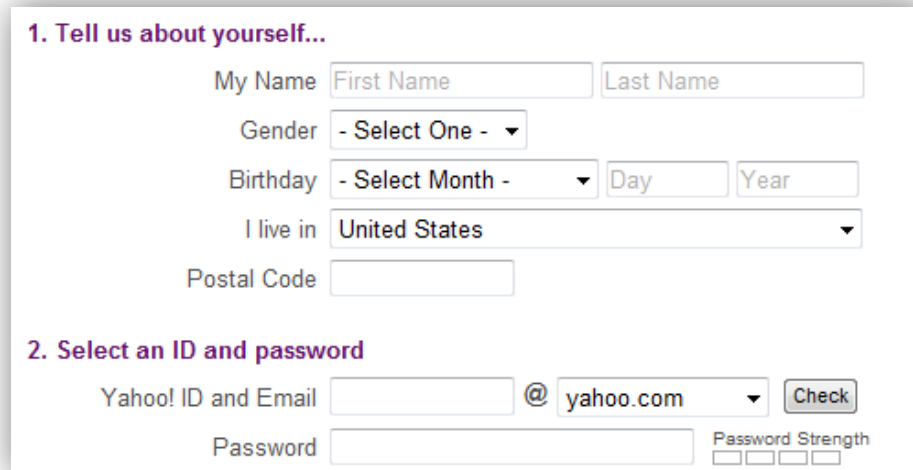
Good Form

The principle of good form refers to the tendency to group together forms of similar shape, pattern, color, etc. Even in cases where two or more forms clearly overlap, the human brain interprets them in a way that allows people to differentiate different patterns and/or shapes. An example would be a pile of presents where a dozen packages of different size and shape are wrapped in just three or so patterns of wrapping paper.

Implications for Designers

This informs our visual design choices when, for example, several blocks of information exist on a page, each with equal import. We are led to style these blocks identically. Conversely, if there is some hierarchical relationship between these blocks, we would style them differently to call out this semantic difference.

Examples



The image shows a user registration form with two distinct sections. The first section, titled "1. Tell us about yourself...", contains several input fields: "My Name" with sub-fields for "First Name" and "Last Name", a "Gender" dropdown menu, a "Birthday" section with a "Month" dropdown, "Day" and "Year" text boxes, an "I live in" dropdown menu set to "United States", and a "Postal Code" text box. The second section, titled "2. Select an ID and password", includes a "Yahoo! ID and Email" text box followed by an "@ yahoo.com" dropdown menu and a "Check" button. Below this is a "Password" text box and a "Password Strength" indicator consisting of four empty square boxes.

Consistency

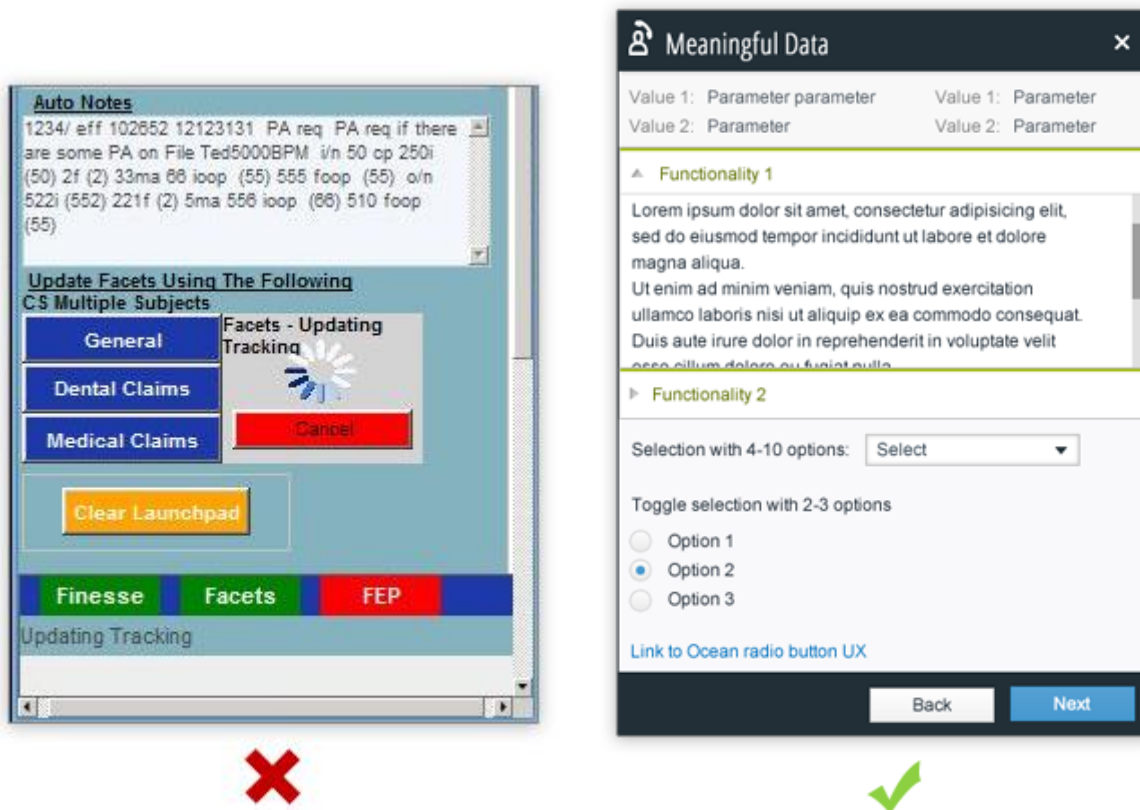
Ralph Waldo Emerson famously said that "A foolish consistency is the hobgoblin of little minds", and while that is true, there is nothing foolish in minimizing the cognitive load on our users. It is by being consistent that we can reduce the amount of brain power required to parse a dialog or perform a certain task. Users are free to learn a pattern of interaction or the semantic coding of a style once and subsequently just concentrate on performing their task. How frustrating it would be if every dialog they used had the OK and Cancel buttons in a different place. How visually distracting would it be if each piece of text on the screen was a different font face, size, style, and color? Parsing all of these inconsistencies wastes cognitive energy.

Implications for Designers

If you have a choice to use an existing template, or copy an existing screen, then do it. If you have to come up with something new, then make it indistinguishable from what already exists. Your touchstone should be the design pattern library available on the wiki.

Examples

Notice how the use of minimum colors, same fonts and graphic elements, create a simple, friendly and navigable dialog. Whereas the use of many strong colors fonts and positions



Miller's Law of Short Term Memory Load

The psychologist George Miller described a variety of phenomenon which suggested that most people can only hold approximately seven chunks of information in their short term memory at once (the Magic Number 5 ± 2). For example, most people can remember a random seven-digit phone number. **Please note that his law does not apply to things like the number of items in a menu or a selection list, because the selection task does not require people to hold the options in memory.**

Implications for Designers

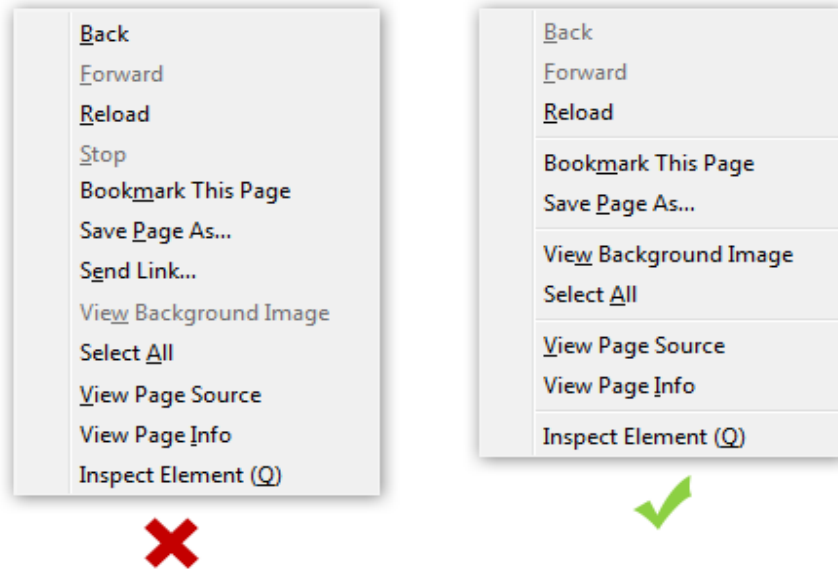
This really means that we should minimize the memory load of our users by having the system remember and present information where and when appropriate. The user should not be forced to remember information presented on one screen to enter it or search for it on a following screen.

Example:

1. We can divide the credit card number (or phone, serial number etc.) the chunks, Hans making it easier to remember and to fill in forms.



2. In the context menu we can divide the commands into sub logical groups to make the dialog easier to scan and more usable to the user.



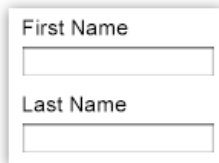
Hick's law **does not apply** if the list is alphabetical and the user knows the name of the command, he or she may be able to use a subdividing strategy that works in logarithmic time.



Alignment Rules

Rules here assume a left-to-right language.

1. Labels relative to their control
 - a. Labels should be above the control if you have sufficient vertical space



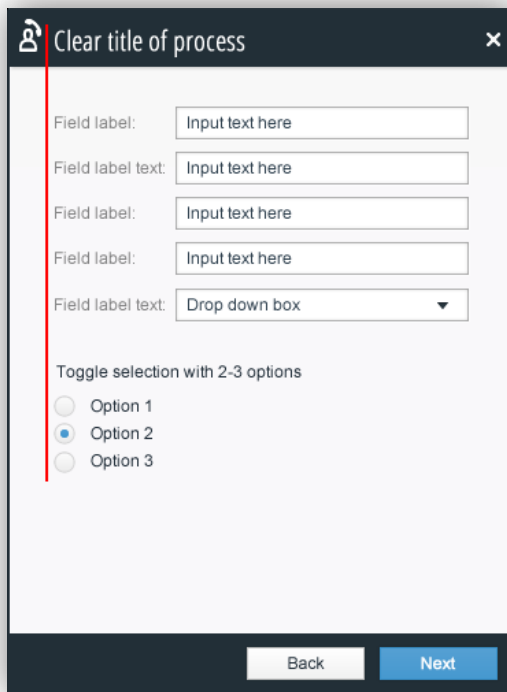
A form with two input fields. The label "First Name" is positioned above the first input field, and the label "Last Name" is positioned above the second input field. This illustrates the rule that labels should be above the control when there is sufficient vertical space.

- b. Labels should be right aligned to the left of the control if vertical space is at a premium













A form with two input fields. The label "First Name" is right-aligned to the left of the first input field, and the label "Last Name" is right-aligned to the left of the second input field. This illustrates the rule that labels should be right-aligned to the left of the control when vertical space is at a premium.

2. Vertical alignment of text, graphics, and controls comprising a line/row should be by the baseline of the first line of text (if there is wrapping)



A dialog box titled "Clear title of process" with a close button (X). The dialog contains several input fields and a toggle selection. A vertical red line is drawn on the left side of the dialog, indicating the baseline for vertical alignment. The input fields are: "Field label: Input text here", "Field label text: Input text here", "Field label: Input text here", "Field label: Input text here", and "Field label text: Drop down box". Below these is a section titled "Toggle selection with 2-3 options" with three radio buttons: "Option 1", "Option 2" (selected), and "Option 3". At the bottom of the dialog are two buttons: "Back" and "Next".

3. Dialog buttons at the bottom of a dialog should right aligned
4. Text in grids
 - a. Heading alignment always matches the alignment of the column values
 - b. Text columns should be left aligned. Text includes alphanumeric strings intended not to be interpreted as quantities, such as a serial number
 - c. Numeric columns should be right aligned

Workstation Login	Role	UUA 12 deploy	UUA 11 deplc ^
Leonard	Agent		
Bill	Supervisor		
	Supervisor		
Kate	Administrator		
Ben	Agent		

5. Page, dialog, and section titles
 - a. These should all be left aligned

References

Much of this has been liberally appropriated from the following sources:

<http://www.measuringusability.com/blog/hci-laws.php>

<http://nanlee.wordpress.com/category/theories-in-hci/>

http://en.wikipedia.org/wiki/Gestalt_psychology

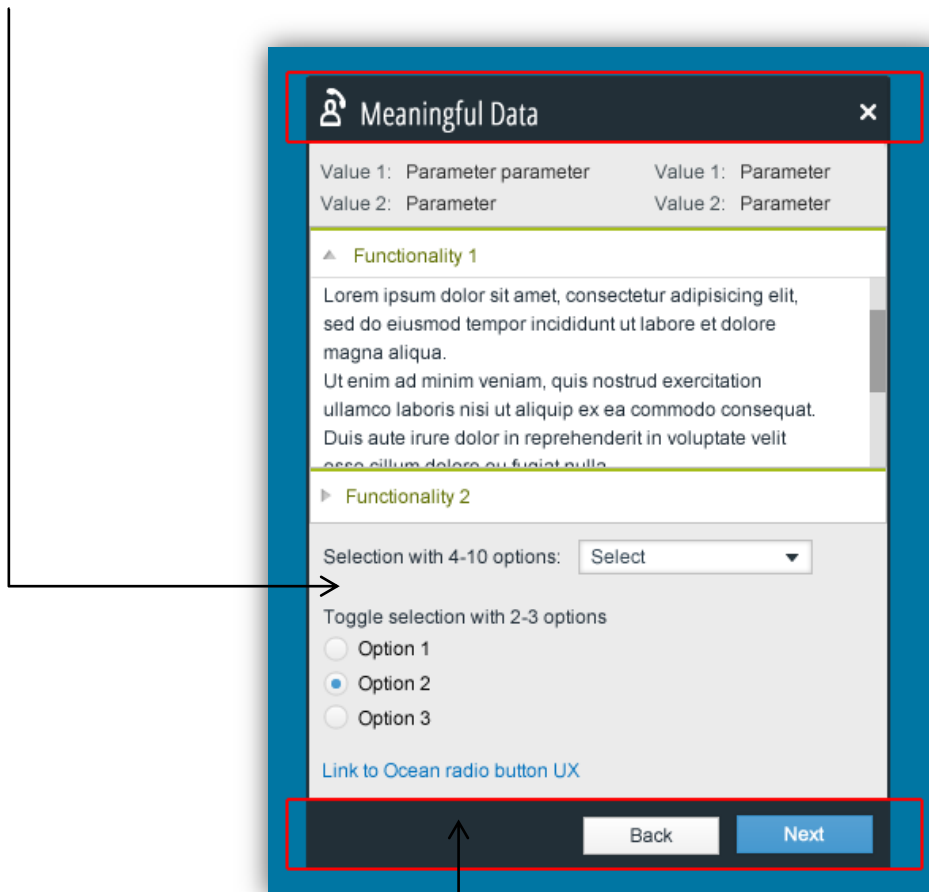
http://en.wikipedia.org/wiki/Principles_of_grouping

Graphic guidelines

Colors

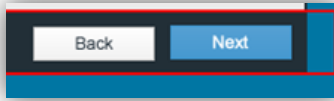
Window background color: #ebebeb

Used Also for Scroll & Slider background



Header & Footer Area: #222f37

Primary (Next) button:



Borders

Regular: border-color:rgb(58,146,203); border-width:1px; border-style:solid;

Hover: border-color:rgb(58,146,203); border-width:1px; border-style:solid;

Press & Focus: border-color:rgb(58,146,203); border-width:1px; border-style:solid;

Disabled: border-color:rgb(223,223,223); border-width:1px; border-style:solid;

Fill Colors

Regular: Gradient (50% 0% -90deg,rgb(86,160,211) 0%,rgb(69,152,207) 100%)

Hover: Gradient (rgb(108,173,217) 0%,rgb(109,174,218) 100%)

Press & Focus: Gradient (rgb(50,134,190) 0%,rgb(43,115,164) 100%)

Secondary (Back) button:

Borders

Regular: border-color:rgb(180,180,180); border-width:1px; border-style:solid

Hover: border-color:rgb(160,190,205); border-width:1px; border-radius:1px; border-style:solid

Press & Focus: border-color:rgb(192,192,192); border-width:1px; border-style:solid

Disabled: border-color:rgb(223,223,223); border-width:1px; border-style:solid

Fill Colors

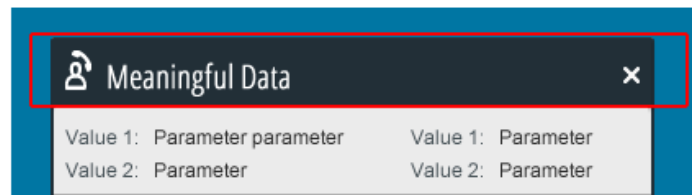
Regular: Gradient (50% 0% -90deg,rgb(255,255,255) 0%,rgb(240,240,240) 100%)

Hover: rgb(209,223,230)

Press & Focus: Gradient (50% 0% -90deg,rgb(245,245,245) 0%,rgb(225,225,225) 100%)

Disable: rgb(246,246,246);

Fonts

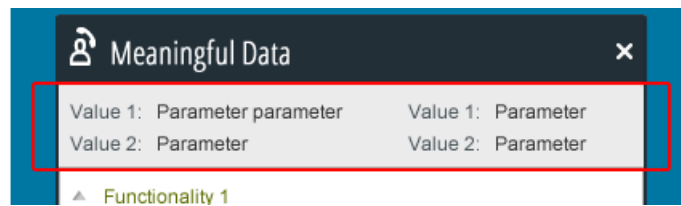


Title for Regular View:

font-family:"Open Sans";font-size:20px;font-stretch:condensed;line-height:120%;font-weight:bold;color:rgb(255,255,255);text-align:left;

Title for Minimized View:

font-family:"Open Sans";font-size:16px;font-stretch:condensed;line-height:120%;font-weight:bold;color:rgb(255,255,255);text-align:left;



Top Information Area

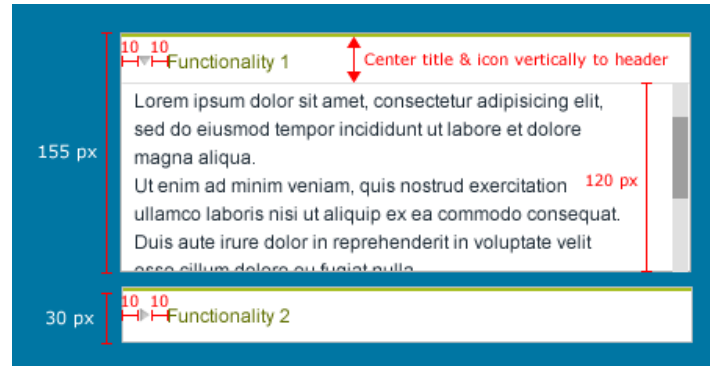
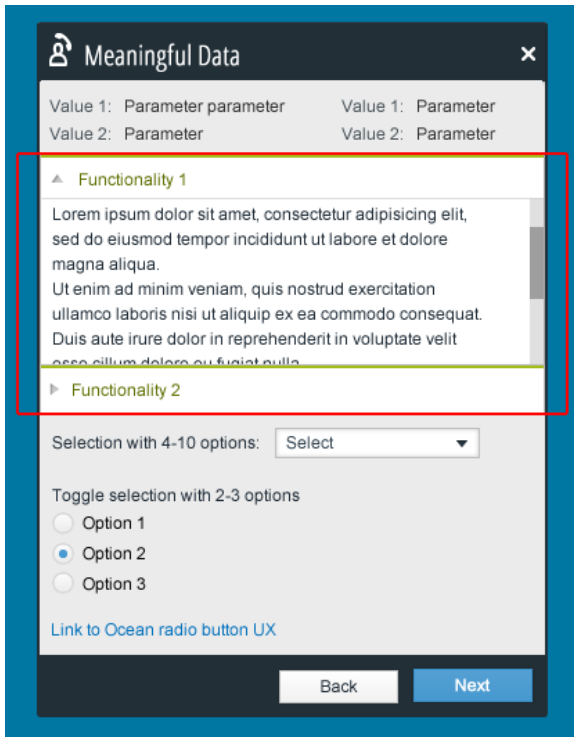
Labels:

font-family:Arial, Helvetica, sans-serif;font-size:12px; //0.75em//line-height:22px;color:rgb(33,46,55);text-align:left;

Values:

font-family:Arial, Helvetica, sans-serif;font-size:12px; //0.75em//line-height:22px;color:rgb(0,0,0);text-align:left;

Panel area



Colors & Borders

Panel Background: #FFFFFF

Panel Border: 1px width #C0C0C0

Top Border: Add 2px tall line #A8C20E

Fonts

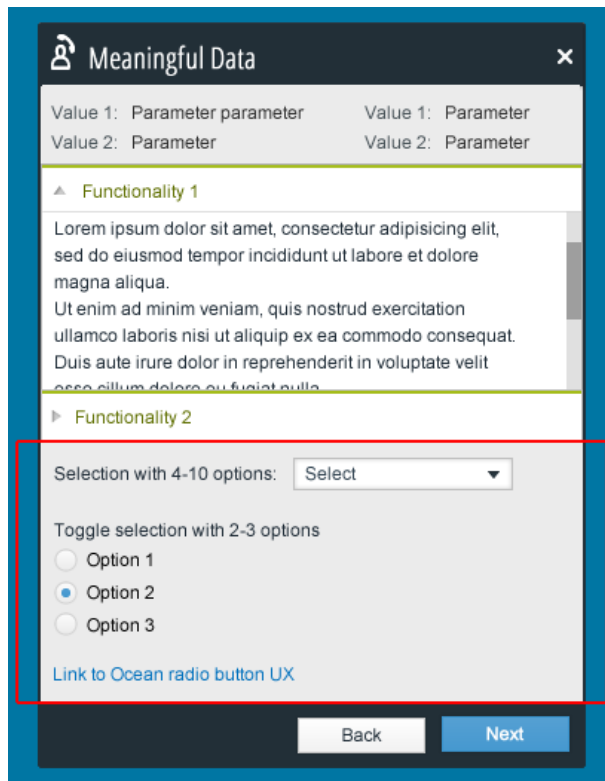
Panel Title:

font-family:Arial, Helvetica, sans-serif; font-size:12px; //0.75em// line-height:22px;color:rgb(96,112,1);text-align:left;

Inside Text:

font-family:Arial, Helvetica, sans-serif;font-size:12px; //0.75em//line-height:22px;color:rgb(33,46,55);text-align:left;

Additional Information & Controls Area



See “controls and how to use them”

Controls

Labels:

font-family:Arial, Helvetica, sans-serif;font-size:12px; //0.75em//line-height:22px;color:rgb(33,46,55);text-align:left;

Values:

font-family:Arial, Helvetica, sans-serif;font-size:12px; //0.75em//line-height:22px;color:rgb(0,0,0);text-align:left;

Links:

font-family:Arial, Helvetica, sans-serif; font-size:12px; //0.75em// line-height:30px; color:rgb(0,113,190); text-align:left;

Reference:

http://192.168.245.94/w/RTPO_graphic_design

Navigation Guidelines

There are many different mechanisms by which a user can get from one place to another within your application. Here are a few things to consider when designing your navigation.

Tabs

It used to be that tabs were reserved for presenting different aspects or sets of attributes for a single object, but today tabs simply delimit logical groupings of functionality or other items that have affinity.

When to Use Tabs?

Tabs are a good choice when most of the following conditions apply:

1. The number of tabs is small. They all need to fit across the page without scrolling
2. The number of tabs is fixed. Allowing a dynamic number of tabs could result in more tabs than fit across the page
3. There is no natural hierarchy to the items you are presenting. Tabs only provide one level of grouping Horizontal menus.

Horizontal Menus

The horizontal menu is a very common form of navigation. The layout is similar to tabs, with a small fixed number of menu items lined up near the top of the page. However, rolling over and/or clicking on the items reveal another layer of information in the form a standard or mega-menu that drops down.

When to Use Horizontal Menus?

Horizontal menus are a good choice when most of the following conditions apply:

1. The number of menu items is small. They all need to fit across the page without scrolling
2. The number of menu items is fixed. Allowing a dynamic number of menu items could result in more menu items than fit across the page.
3. There is at most a very shallow natural hierarchy to the items you are presenting. Menu items only provide one level of grouping, but the menus that drop down provide a second level. If you are using mega-menus, then you can have groups of items on the dropped down menu, essentially yielding a third level Trees.

Trees

Trees usually appear on the leading edge of the page (the left side for Western Languages). They commonly allow for large numbers of items and provide vertical scrolling independent of the main content window next to them when their size demands it. The tree can support almost any reasonable number of child levels. Horizontal scrolling of the tree is sometimes supported, as is manually resizing the width of the tree, or collapsing it completely.

When to Use Trees?

Trees are a good choice when any of the following conditions apply:

1. The number of menu items is large. They can all be displayed via vertical scrolling
2. The number of menu items is dynamic. They can all be displayed via vertical scrolling
3. There is a natural hierarchy to the items you are presenting, and it may be deep. Trees provide great flexibility with grouping and the number of sub-groups that are possible Drill Throughs.

Often summary information will naturally lead to curiosity about the details behind it. The user would like to drill through to the details. This is usually available by clicking on some element of the summary. The detail information may be presented as a page that replaces the launch point page, or it may be popped up into a dialog that obscures most of the screen. Regardless of how the detail information is presented, it is crucial that a clear path back to the launching point exists.

Drill Throughs?

When to Use Drill Throughs?

Drill throughs are a good choice when most of the following conditions apply:

1. There is a clear summary-detail relationship
2. There is a very short chain summary-detail relationships possible. Too many levels is hard for the user to keep track of without an explicit Breadcrumb Trail.
3. There is a clear path back up the chain to each of the launch points.

Direct Links

Direct links can be used similarly to tabs or drill throughs, or used for any other navigation path directly to something related to the current screen. The destination of these links may be some far removed branch of your application and thus may be very disorienting to the user. It is therefore very important to make clear to the user where they are about to go before they click on the direct link, and make it very clear where they have landed, with a clear link back to where they came.

When to Use Direct Links?

Direct Links are a good choice when most of the following conditions apply:

1. There is no other appropriate navigation system already in place
2. The information represented by the link has close affinity to other items on the screen and the link is located physically near to those other item
3. There is a clear path from the destination page back to the launch point

Breadcrumb Trails

Not a navigation strategy on its own, but is a useful mechanism for answering the user's question "where am I?" and "how did I get here?", and for allowing the user to jump back to a place they have recently visited.

Usability testing

What it is

Usability testing is a technique used to evaluate a product by testing it on users. It focuses on measuring a product's capacity to meet its intended purpose.

What it is NOT

Gathering opinions on a feature or product.

Usability testing is a systematic observation under controlled conditions to determine how well people can use the product. Rather than showing users a rough draft and asking, "Do you understand this?", usability testing involves watching people trying to *use* something for its intended purpose.

When to do it

Usability testing should take place throughout the design/development process, to detect problems or potential interaction failures with the product.

By doing the testing early enough you will be able to catch and correct problems before they are implemented.

How it is done

- a session should schedule 3-5 participants. This is usually enough to find most of the serious usability issues, and this is the amount that can be comfortably scheduled in one morning.
- The participants do not need to be real users of the system, but they should reflect the background and skills of real users.
- Make a list of 5-10 of the most important tasks that users need to do with the system.
- Decide on which ones the test is based on:
 - What are the most critical tasks? The tasks that people must be able to do in order to reach their goals.
 - What items do you think may have usability problems? If there are things in the application that you already suspect as problematic then they should be tested. What items have already received negative feedback?

Create a scenario

After selecting the tasks to be tested you will need to create scenarios based on those tasks for the users to attempt to follow.

The scenarios should reflect the real use of the users of the system and fulfill the customer goals.

Running the test (“Hallway testing”)

1. Invite a user to sit in front of your system.
Note- it does not have to be a real live system.
Usability testing could be done with a pencil and a paper (Paper Prototyping), as long as the flow and structure are the same as in a real product. (see http://en.wikipedia.org/wiki/Paper_prototyping)
2. Give the user a **short** explanation about the system.
3. Give the user a task/scenario and ask him to follow it.
4. Have him think aloud and describe what he’s trying to do.
5. **Do not comment nor help** or give him hints unless he fails completely and get stuck and frustrated (write that down as a UX fail)
6. At the end of each session, have the users explain the 3 most serious usability problems they saw.

Your goals are:

1. Tell the user what to do and keep them moving forward.
2. To get them to verbalize their thoughts while they use the system.
3. Listen, Watch, learn and take notes of bugs and problems.

Controls

Controls, sometimes called “widgets”, are the individual data collection and display elements that you put on your forms and dialogs. Each one comes with its own set of usage guidelines. They are collected here, below.

Check Boxes

Formatting

- Ignore colors specified on webpages
- Ignore font styles specified on webpages
- Ignore font sizes specified on webpages

A check box appears as a square box with an accompanying label. When the choice is selected, a check mark appears in the box. When the choice is not selected, the check box is empty. A check box can have one of three states:

- Checked — the associated value is set.
- Cleared — the associated value is not set.
- Mixed value — the associated value is set for some, but not all, items in the current selection.

When to Use Check Boxes

Use a check box only when it provides choices that are opposite and unambiguous. Otherwise, use option buttons.

On a New Blank Form

On a new blank form, the check box should be in the default state, whatever that may be for the circumstance.

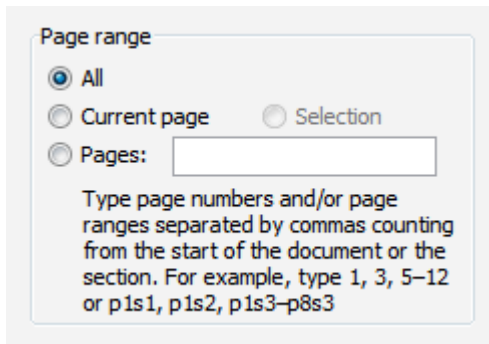
Do

- Place the label to the right (for left-to-right languages) of the check box, and have the label be part of the click target
- The box should have a square shape.
- When the choice is set then the box will be marked, and when the choice is not set then the box will be empty.
- A 'mixed state' indication can be used if the check box is related to a group of partially selected items.
- Clicking anywhere on the square or the text should toggle the item.
- Related check boxes should be grouped.
- If the input focus is on the check box, then pressing the keyboard 'Space' key will toggle the value.

Don't

- When both states of the choice are not clearly opposite and unambiguous.
- To start an action (Primary or Regular buttons should be used instead).
- When there are too many options and not enough space for them.

Radio Buttons



Page range

All

Current page Selection

Pages:

Type page numbers and/or page ranges separated by commas counting from the start of the document or the section. For example, type 1, 3, 5-12 or p1s1, p1s2, p1s3-p8s3

An option button is a control that represents a single choice within a limited set of mutually exclusive choices. That is, the user can choose only one of a series of options. Option buttons are sometimes called radio buttons because they function like the presets on old-fashioned mechanical radios, in which pushing buttons moves the needle that indicates the selected channel. Similarly, the user's selection of an option button is indicated by a dot appearing in the button.

When to Use Option Buttons

Use option buttons only when they provide mutually exclusive choices and when there are only a few choices. When there are many choices, use list boxes instead.

- To select one item from a group of 2-7 items
- When it is important to see all of the options quickly (without having to open a drop-down or scroll a list).
- When there are very long labels for each item.
- When there is enough room to show all of the items.
- In situations where the alternatives cannot always be easily remembered.
- Where displaying the alternatives together facilitates understanding and selecting the proper choice.

On a New Blank Form

On a new blank form, the default option button should be selected, and the others not selected. In rare exceptions, namely, Likert scale survey forms, no option button may be the starting state. Otherwise one option button in the group must always be selected.

Do

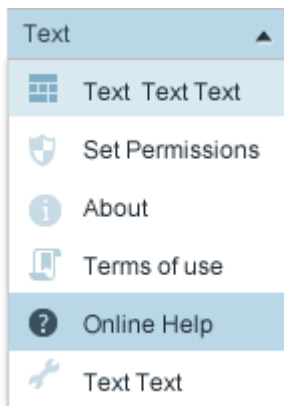
- Place the label to the right (for left-to-right languages) of the check box, and have the label be part of the click target
- Should have a round (circle) shape.
- When the item is selected there should be a dot within the circle, and when the item is not selected the circle should be empty.

- A 'mixed state' indication can be used if the radio buttons are related to a group of partially selected items (e.g. when editing a setting for a group of selected people, the initial state for each person might be different so the Radio Buttons will show this by using the mixed state indication).
- Clicking anywhere on the circle or the text should select the item.
- There should be an indication as to what the boundaries of the group are (which items belong to the group).
- The label text and the order of the items should not change dynamically (i.e. based on the state of the system).
- Usually there should be a default selected value (however, if you want to be sure the user really looked at the question, then not having a default will force them to make a selection).

Don't

- When there are more than seven items in the group (Drop-downs, Lists or Pickers should be used instead).
- To start an action (Primary or Regular buttons should be used instead).
- To toggle between two opposite states (a Check Box (UX) or Button-Toggle (UX) should be used instead).

List Boxes



List boxes are basic rectangular boxes in which the list of values is always visible. These lists may be scrolling or not. While most allow just one item to be selected, some allow multiple selections.

The items in the list appear in alphabetic order unless some compelling contextual reason suggests some better ordering.

When to List Boxes

Use a list box to when there is enough space for the box or when multiple selections are required. When activated, the user can “blind type” and the list scrolls to the nearest match.

- For selection of items or setting attributes when there are more than 6 items to choose from.
- The width should be enough to display the longest item.
- When it is important to see many of the options immediately and at the same time.
- When the list contains items from a database (radio buttons and check boxes are not appropriate for this).

On a New Blank Form

On a new blank form, the default list item should be selected. If having no selection is a valid state for the completed form, then a blank row must be the first choice in the list and should be the default.

Do

- Should be clear to the user whether it is single-select or multi-select. The label should help make this distinction (e.g. "please select one item" vs. "select all relevant items").
- The size of the box should be enough to display at least 5 items at one time.
- The items should be ordered in a logical manner (e.g. based on frequency or alphabetically).
- A vertical scroll-bar should be added only if there are more items than can be shown in the box at once.
- Selecting the item (i.e. clicking on it) should not activate the process. There should be an explicit action (e.g. pressing a regular or primary button).
- The list can display items in a hierarchical way. In case the data is represented in a hierarchical way, each list item can include
 - underneath several levels of hierarchy and the user can expand or collapse the view of each level by an appropriate control (+/-).
 - Use the hierarchical list when the hierarchy have less than three levels, for more than that use the tree view control.

Don't

- When space is limited (you can use Drop Down List (UX) instead).
- When the hierarchy have more than 2 levels.
- When the control is not used for selection only, but to extended operation of the system, use a List

Drop-Down List Boxes



Drop-down list boxes display the current selection and hide the list of choices until activated.

The items in the list appear in alphabetic order unless some compelling contextual reason suggests some better ordering.

When to Drop-Down List Boxes

Use a drop-down list box when space is constrained and only a single selection is allowed.

- When there is not enough space to show all the options.
- When the number of options is greater than 7.
- When it is not important to show all the options all of the time.

On a New Blank Form

On a new blank form, the default list item should be selected. If having no selection is a valid state for the completed form, then a blank row must be the first choice in the list and should be the default.

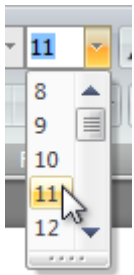
Do

- After selecting an item, the selected item will then appear in the text box.
- Clicking anywhere on the text box or the button will open the drop-down.
- The text box width should be large enough to show the largest item in the list.
- The list height should be enough to hold from 3 to 8 items.
- If the list contains too many options then it should accommodate scrolling.
- When the list is open then clicking on an item in the list will select that item and close the list.
- When the list is open, if the user presses the ESC key or clicks anywhere outside of the control then the list will close and no changes will be made.
- If it is a single-select pull down then each selection made will replace the previous selected item.
- If it is a multi-select pull down then each selection made will be added to the list of selected item.
-

Don't

- When the items have very long names.
- When it is important to compare all of the options or expose all options to the user.

Drop-Down Combo List Boxes (i.e. Combo Boxes)



A drop-down combo list box combines the characteristics of a text box with those of a drop-down list box. The text box and its associated list box have a dependent relationship. As the user types text in the box, the list scrolls to the nearest match. If there is no exact match, then the entered text becomes a new value in the list, and the current selection as well.

The items in the list appear in alphabetic order unless some compelling contextual reason suggests some better ordering.

When to Drop-Down Combo List Boxes

Use drop-down combo list boxes in place of option buttons to conserve space and when the user needs to add entries to the list.

- To enable to select from a list and typing a value directly into the text box.
- For short items (such as numbers) that have a specific format
- When you can validate the data entered

On a New Blank Form

On a new blank form, the default list item should be selected. If having no selection is a valid state for the completed form, then a blank row must be the first choice in the list and should be the default.

Do

- List the items in an intuitive order.
- Default to alphabetic or numeric order for items that are not specially ordered.
- If "None" is a valid value, then include "None" as a choice in the drop-down menu.
- Consider adding user-entered items to the menu. An example is a menu of IP addresses that users might want to visit again.

Don't

- When you can't validate the data entered
- For a very short list of options

Command Buttons



Users click command buttons in order to immediately perform an action, such as committing a change or viewing detailed information. A command button is rectangular and includes a label. Command buttons appear in a variety of contexts. A given property sheet, for example, might have command buttons that affect a specific text box, the active tab (page), or all the tabs. The location of the command button determines its relationship to other controls.

When to Command Buttons

The command button control causes an application to perform an action when the button is selected.

- To initiate an immediate action.
- To display another window (or pop-up).
- To display a menu of options.

Do

- The button should look and to communicate as clickable.
- The effect of mouse over or pressing a button should be immediate.
- The label of the button should be short and should clearly describe the action the button takes.
- A button that leads to a cascading dialog should include an ellipsis (...) after the label.
- A button that causes the dialog window to expand should include some visual indication as a double arrow (>>) pointing in the direction that the dialog will expand.

Don't

- Setting individual parameters (use Radio Button, Button-Toggle or Check Box instead).

- Setting options that affect the entire application (should be on menus or toolbar).

Date Pickers

Time period

5/1/2013

The date picker control appears like a drop-down combo list box in its resting state. It provides a drop-down calendar display that enables a user to select a date and display it in a specified format.

Date pickers must always allow the manual entry of dates by typing or pasting in the text area.

When to Date Pickers

1. Use this pattern when the user wants to easily apply a date to an item. This includes contexts where the application itself needs to apply date metadata ("date-stamp") a piece of information in order to track, filter, and sort items by date or date order.
2. Common scenarios include events, tasks, and appointments, as well as items usefully displayed in chronological or reverse-chronological order, such as blogs, photo uploads, and email messages.

Do

- The calendar picker is activated by clicking a calendar icon most often placed next to the field used for inputting a date.
- On activation, a box with a month-calendar is laid over the current page, prompting the user to select a date in the box. It is most common to only show one month, but some interfaces show up to 3 month calendars next to each other to ease the click-burden of the user and provide a better overview.
- The date picker detailed behavior, navigation, and operation should follow [Kendo ui](#)
- Usually you should allow the user to input the date also via the text field and not force him to select from the calendar control. Keep in mind that this will require validation of the entered value.
- The defaults to pre-select depend on the context, but will most often be the current date or time.
- Fit the control to the task. If the task require selecting only a month or only a year then customize the control to accommodate this.

Don't

Do not use in isolation, when the user is more familiar and efficient with another way of inputting a date. Some users prefer inputting a date via a text field.

Text Boxes

Regular:

A text box (sometimes called an edit control) is a rectangular control in which the user types (or pastes) text to interact with the computer. A text box can be defined to support single or multiple lines of text.

When to Text Boxes

The text box control provides basic user text input and editing support.

On a New Blank Form

On a new blank form, the text box is usually blank. If an expected default exists, like a known attribute of the current user, for example, then it should be used.

Do

- Try to match the length of the field to the expected length of text to be entered, but also keep in mind the length of nearby controls and adjust the length of your text boxes to create nicely rectangular groupings.

Don't

- When the data to be entered is bounded to a set of items (use a list or drop-down list instead).
- For running commands.

Spinners

With a spin control, users can click arrow buttons to change incrementally the value within its associated numeric text box. The term spin box refers to the combination of a text box and its associated spin control.

When to spinner

- For setting attributes, properties, or values that are mutually exclusive.
- When some users know exactly what to enter (they can enter it directly into the field) and others need to be reminded (they will use the up/down buttons).
- When screen space is limited.
- When the set of items is not large (numbers under 100)
- When the set of values are incremental numbers

Do

- Always pair the control with a text box
- Use spin boxes only for exact, known numeric values.
- For contiguous value range only - If not, use a drop-down list
- If the value has delimiters, the associated text box should have multiple input focus points

Don't

- Using a spin control should never result in an incorrect value
-

References

Much of this has been liberally appropriated from the following sources:

<http://msdn.microsoft.com/en-us/library/bb158625.aspx>

[http://msdn.microsoft.com/en-us/library/windows/desktop/bb246433\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb246433(v=vs.85).aspx)

<http://192.168.245.94/w/Category:UX design patterns>

UX Do's and Don'ts – quick tips

Good UX

- Requires less mental effort to use.
- Less mistakes when using it.
- Do more, with less (steps, clicks, etc).
- Intuitive: Easy to learn/use & familiar to user.

General

- Embrace the KISS (keep it simple stupid) approach.
(http://en.wikipedia.org/wiki/KISS_principle)
- Try to keep everything **not more than 3 clicks away**.
- User should not wait **more than 2 seconds** for any response
- Main goals of good UX: **Simple & elegant, clear & consistent**.
- Try to **break long and complex procedures** to short and clear steps
- Don't leave users in The Dark: **keep the user informed of what is happening**. Use loading detailed indicators, tooltips and informative labeling
- Avoid using the words "click here" for a button. Buttons should always be **descriptive** text.
- Use **meaningful title bars** to summarize callout content. Keep titles under 75 characters.
- **Error prevention**- Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
- **Recognition rather than recall**- Minimize the user's memory load by making objects, actions, and options visible.
- **Error messages** should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- **Keep menus short** - A shorter menu with distinct categories is easier for mobile visitors to navigate.
- **Minimize form errors** with labeling and real-time validation - Use clearly visible labels to let users know what you need, and validate for errors in real time to let them know if there's a problem before they submit a form.
- **Design efficient forms** - Minimize the number of fields in your forms, and auto fill information wherever possible. Use clearly-labeled progress bars to help users get through multi-part forms.
- Use **Confirmation dialogs**: Help the user understand what's going on.

Text

- **Text should be legible** at a typical viewing distance without zooming.
- Make sure there is **sufficient contrast** between the text color and the background. This will improve readability.
- **Don't let text overlap**. Improve legibility by increasing line height or letter spacing

- **Do Not cram** more into a space than what can fit comfortably
- **Avoid using all caps.** It is easiest to read text that is in the standard upper and lower case format.
- **Labels inside input fields:** give the user hints and guides of what is accepted (e.g. "6-12 characters that contains numbers")
- **Choose a simple and common font:** *fancy fonts* are not readable and make the user work hard to decrypt.
- Define a set of up to 3 font sizes and stay consistent with them throughout the application.

Graphics

- **Do not overuse icons:** Keep the callout clear and use clear and simple icon for the each logical procedure.
- Do not make the user jump through hoops to fill out a long form, **keep it simple and minimal** and if possible, break it down to a wizard.
- Use color and contrast to shift focus to important items but be careful to not overdo it.
- Always use **brand colors and graphics**, unless redesign and customization is a project definition. In this case try to keep the customization minimal.
- Don't use color as the only visual identifier, 4-10% of the population are colorblind...
- **Convey meaning with colors:** e.g. Red=Alert, Yellow=Warning, Green=Approve/Success etc.