

# Design: Interaction



# Design Interaction

---

## Table of Contents

- About this Guide .....4**
- Introduction.....4**
  - Why Design Matters ..... 4**
  - Why Design is Important to Learn ..... 4**
- Make it Easy to Learn .....5**
  - Design your interface so it explains itself. .... 5**
    - Use familiar patterns ..... 5
    - Know your users ..... 6
    - Use visual formatting to help users differentiate interactive and non-interactive elements..... 6
    - Use the minimal visual formatting necessary to signal interactivity ..... 7
    - Apply styles and themes to ensure the consistent visual formatting..... 8
    - Pair icons with labels – at least at first ..... 9
  - Provide Clear and Responsive Feedback ..... 10**
    - Provide feedback about user actions ..... 10
    - Provide feedback about scripts and processes that the user must wait to complete ..... 12
    - Integrate status information into the interface ..... 13
  - Provide navigation elements to orient and direct your users..... 14**
    - Define the hierarchy of data and functions ..... 14
    - Arrange and design layout element styles to mirror the underlying hierarchy ..... 15
    - Provide landmarks to help orient users..... 17
    - Clearly label navigational elements ..... 18
- Make it Efficient to Use .....20**
  - Focus interfaces to help users achieve specific goals ..... 20**
    - Design for specific scenarios ..... 20
    - Provide a clear path to completion..... 21
    - Delete or move secondary content ..... 23
  - Be Consistent and Monotonous..... 24**
    - Be consistent in the placement, look, and function of elements ..... 24
    - Be monotonous ..... 26
  - Don't require great precision ..... 28**
    - Don't size targets smaller than the minimum recommended size ..... 28
    - Maximize target size ..... 28
    - Minimize the distance to the target ..... 29
  - Support the Development of Productive Habits..... 30**
    - Set a logical tab order ..... 30
    - Set field exit options ..... 30
    - Enable the use of keyboard shortcuts..... 31
    - Be consistent in how you configure custom dialogs ..... 32
  - Anticipate, Prevent, and Help Users Recover from Errors..... 33**
    - Identify potential errors ..... 33

Highlight required fields .....	34
Constrain choices .....	34
Provide inline validation .....	35
Provide actionable error messages .....	35
<b>Don't interrupt</b> .....	<b>37</b>
Modal windows .....	37
Custom dialogs .....	38

---

## About this Guide

This guide outlines best practices for designing FileMaker solutions to provide an optimal user experience. Solutions that achieve an optimal user experience provide a balance between making it easy for new users to learn the solution and making it possible for intermediate and power users to work efficiently.

### Who can benefit from the guide?

This guide is designed for all levels of FileMaker developers who want to create well-designed solutions. Novice developers will find specific actionable steps to improve the design of their solutions. Experienced developers can use the content to ensure they are applying interaction design best practices.

### What you will learn

You will learn best practices of interaction design and specific ways to apply those practices when designing solutions with FileMaker Pro. This guide will focus specifically on:

1. Designing solutions to enable new users to learn through exploration and recognition.
2. Designing solutions to enable users to be as efficient as possible.

## Introduction

### Why Design Matters

A well-designed solution can help a person or company achieve their goals, solve previously insurmountable problems, and accomplish business tasks.

However, if your applications are poorly designed, users may waste significant time and money on repetitive and inefficient processes, and feel frustrated, annoyed, or helpless.

### Why Design is Important to Learn

While you might believe the software you design is easy to use, your users may not share your view. As the designer of the system, your solution will always be more intuitive and easier for you to use. After all, you have a model for how your application works in your mind, you know what each button does, you understand the placement and function of each data element, and you know all the shortcuts.

Other people are not so fortunate. They don't have access to your knowledge about how you designed the system, so they have to learn to use the solution through exploration and training. In short, if you want others to easily learn to use your solution, your interfaces must clearly communicate how they should be used.

The following technical brief will guide you through the key principles of interaction design, and it will give you simple strategies to create better FileMaker Pro interfaces.

## Make it Easy to Learn

Software is easy to learn when new users can look at an interface and know immediately what to do based on past experiences and current cues. Your objective would be for a new user to navigate your interfaces through exploration and recognition.

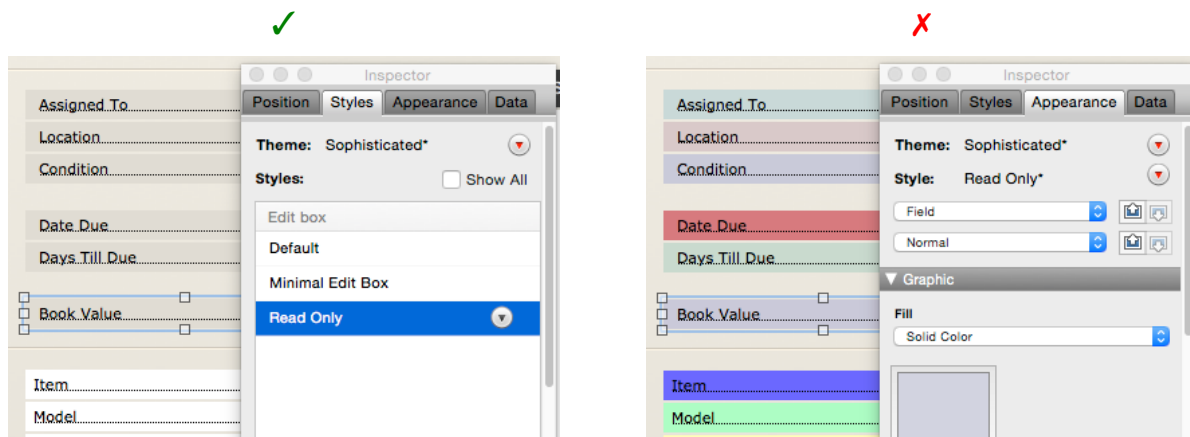
People create internal mental models to explain the workings of the world, other people, and events. Based on what we can see and the responses to what we do, these models help us adapt to new situations and re-use the things we re-encounter.

You can help people learn your application by:

1. Designing your interface so it explains itself
2. Providing clear and responsive feedback
3. Providing navigation elements to orient and direct users

## Design your interface so it explains itself

*Use familiar patterns and styling to signify how elements should be used*



### Use familiar patterns

Software is easiest to learn when users can recognize elements they have used before in other applications – and when those elements function as expected. Each interaction or UI element is called a “pattern”.

A person that has only ever driven a car built in 1995 can still get into a car built in 2015 and drive it. This is because their mental model of how a car should work is mostly the same. The steering wheel is round and on the same side, the pedals are placed in a similar location, the gear is within a similar reach, etc. The person doesn't need to relearn very much because both cars follow similar patterns. The same is true for interface design.

It is important to pay attention to these patterns in all applications you use and most importantly, the applications your users use. Even the smallest details can be considered a pattern. For example, when pressing “tab” takes you to the next field in a form, pressing enter submits things, etc.

To insure that users are not misled, don't misuse common patterns by using controls differently than they would expect based on their prior learning. For example, if they learned that a rounded rectangle with blue background and white text is a button, that style should not be used on a field.

## Know your users

Never assume everyone has the same mental model as you. If your users are used to working on Excel, they understand what most of the icons mean, they understand how to navigate sheets, how to enter data and calculations, etc. Similarly, on the flip side, if your user has never used Adobe Photoshop or an image editing software they might not be familiar with the concept of layers, toolbars, etc.

The patterns your users will be familiar with are often established by the platform and the applications they use most. For example, the mobile world has brought a lot of new patterns to user interface design. The more widespread mobile usage becomes, the more people will be familiar with those patterns and expect them. If your solution will be used on mobile devices you may want to consider adding some of those common patterns to make the application feel more intuitive.

The OS X Human Interface Guidelines, iOS Human Interface Guidelines, and Windows Interface Guidelines are recommended reading for understanding the design patterns users will frequently encounter on each platform. These documents are helpful, but it's still important that you do research and find out as much as you can about your users. Generally speaking, you must take into account the user's age, the platform they use, what software they use, what websites they visit, what mobile device they have, etc. A solution that works well for a college student or a tech savvy person may not work well for a person with little experience with computers.

OS X Human Interface Guidelines

<https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/OSXHIGuidelines/>

iOS Human Interface Guidelines

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>

Windows Interface Guidelines

<https://msdn.microsoft.com/en-us/windows/desktop/aa511258.aspx>

## Use visual formatting to help users differentiate interactive and non-interactive elements

The color, line, fill, shadow, icons, text formatting, and positioning of an object all help a user know they can interact with that object. Interactive or editable elements should be formatted in a way that invites interaction. The visual language you use to differentiate interactive objects from non-interactive objects should be applied consistently across the solution.

For example, in Figure 1, the fill gradient, lines, corner radiused, and icons all signal to users that the fields are editable. In contrast, in Figure 2, the absence of fill, gradients, border lines, corner radiused and icons would signify to users that the fields were not edible and could lead to the fields being left empty or unedited.

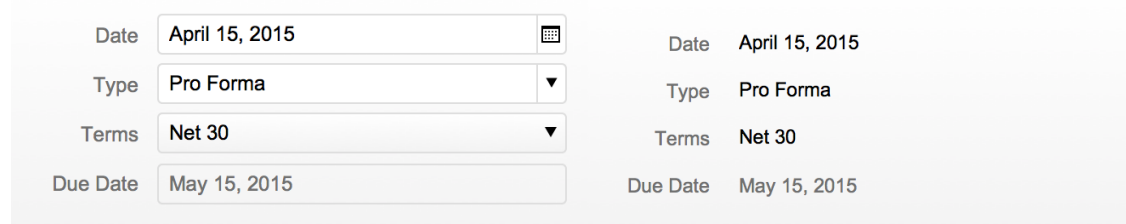


Figure 1

Figure 2

Elements that are only for display, review, or reporting should be formatted so they do not look editable or interactive. For example, while each of the **List View** rows in Figure 3 are a clickable button, the number, name, and date fields are not editable. Imagine how confused you would feel if you clicked into the company name field in Figure 4, and instead of being able to edit the company name you were immediately switched to another layout.

Showing 6 Results		
Unpaid Overdue		
5	Company Name	9/15/2014
4	ABC Consolidated	9/15/2014
3	Company Name	8/5/2014
2	Company Name	7/4/2014
1	ABC Consolidated	6/30/2014

Figure 3

Showing 6 Results		
Unpaid Overdue		
5	Company Name	9/15/2014
4	ABC Consolidated	9/15/2014
3	Company Name	8/5/2014
2	Company Name	7/4/2014
1	ABC Consolidated	6/30/2014

Figure 4

### Use the minimal visual formatting necessary to signal interactivity

While software designers used to rely on a heavy combination of color, gradients, and shadows to simulate three-dimensional interactive objects and indicate interactivity, modern design trends are moving toward using more subtle cues. Users now have significant experience with software applications and digital devices, so it is no longer necessary to absolutely mimic the three dimensionality of the real world on two-dimensional screens.

The InqPad application designed by The Proof Group (Figure 5) is a nice example of the contemporary graphic style. In the screenshot below, the designer has used no gradients or lines to denote that **Inquire** and **Learn More** are buttons. Instead, the fill color, font size and weight, and ellipsis on **Inquire**, and the right-facing arrow next to **Learn More**, are enough to signal to users that the gray regions at the bottom of the page are interactive buttons.

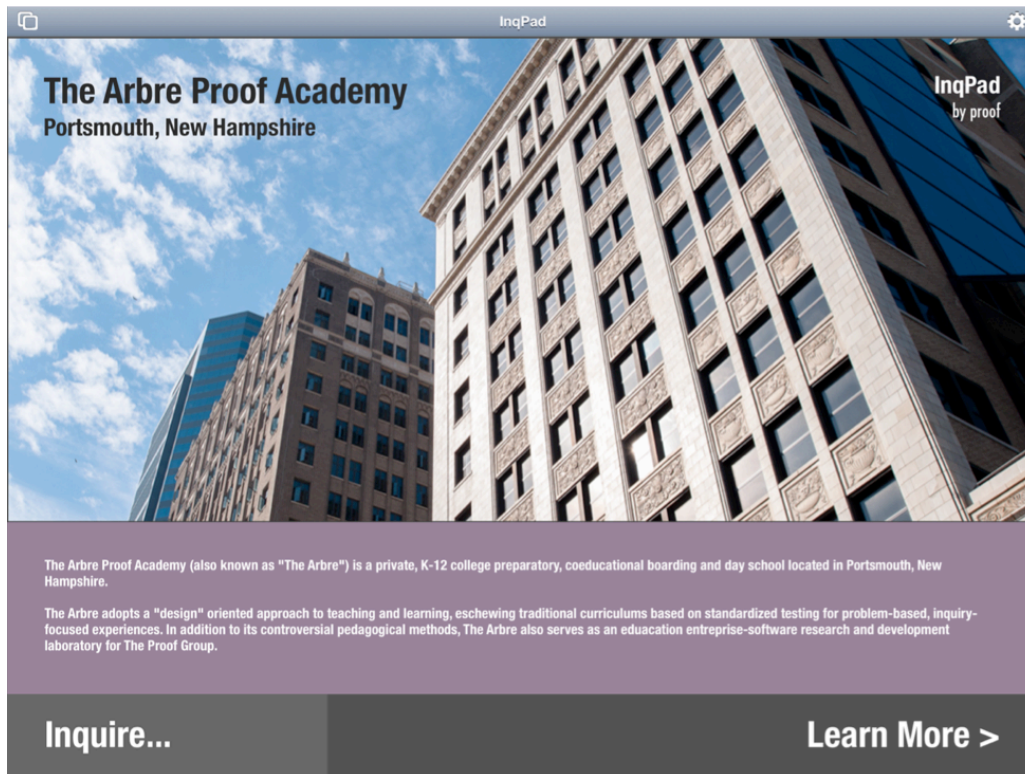


Figure 5

### Apply styles and themes to ensure the consistent visual formatting

FileMaker provides many built-in themes to help you develop and apply consistent visual formatting. Each **Theme** has a number of **Styles** defined for each element type. To apply a style (Figure 6)

1. Select an object.
2. Open the **Inspector** to the **Styles** tab.
3. Select the style to apply.

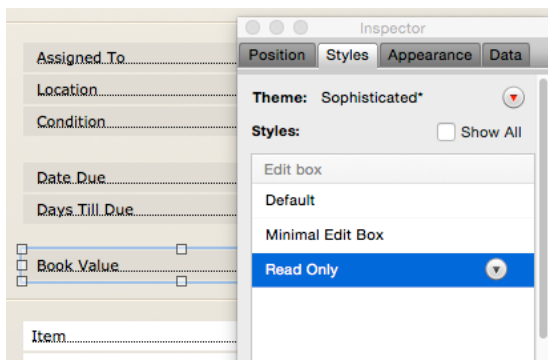


Figure 6

Read this article for more information about using **Themes** and **Styles**:

[http://help.filemaker.com/app/answers/detail/a\\_id/11895/~/-/introduction-to-styles-and-themes-in-filemaker-pro](http://help.filemaker.com/app/answers/detail/a_id/11895/~/-/introduction-to-styles-and-themes-in-filemaker-pro)



### Pair icons with labels – at least at first

A judicious use of icons can aid in learning through recognition, increase scannability, and provide clear tap targets. However, overuse of icons can cause an interface to be noisy and overwhelming. In addition, there is a small set of icons that are truly iconic and universally understood. You can use icons to help your users learn by:

- pairing most icons with labels – especially on the first interfaces in which the user is going to first encounter the icon.
- ensuring that the icons that are presented without labels are truly iconic – and instantly recognizable, or have been presented earlier in the user’s interactions with the solution (so that their function has already been learned).

For example, Figure 7 depicts the bottom toolbar of a FileMaker Go solution designed for mobile service technicians. Pairing the icons with text allows the technician to utilize recognition when using the interface. Were the text labels to be omitted, the technician would need to press each icon to learn it’s use.



Figure 7

### Quick Tips

#### Do

- Draw on familiar patterns to enable users to use recognition for learning your solution.
- Format interactive objects in a way that invites the user to interact with them.
- Format elements that are only for display, review, or reporting so that they do not look editable or interactive.
- Take advantage of the built in styles and themes.
- Use visual styling options and icons to signal interactivity
- Pair icons with text labels.

#### Don't

- Format non-interactive elements to look interactive.
- Use familiar controls in a manner different than users would expect based on their prior learning.
- Overuse icons.

## Provide Clear and Responsive Feedback

Clear and responsive feedback helps the user learn and trust your software.



### Provide feedback about user actions

Feedback is the information that your application sends back to users about the result of their actions and the state of the system. Responsiveness is the speed at which your solution responds, after a user's action or input. Feedback and responsiveness together help reinforce user's expectation of what the application is and does and creates a sense of trust between the user and your software.

FileMaker Pro and FileMaker Go provide many tools you can use to give feedback to your users. These include the **Object-Display-State** styling (**Normal**, **Hover**, **Pressed**, and **Focus**), **Conditional Formatting**, **Merge Variables**, new windows, and **Custom Dialogs**. Always use the least obtrusive method possible while giving the user the information they need.

The **Object Display State** styling settings are designed to help you provide visual feedback to users about the actions they have taken. There are four states you can style:

- **Normal:** The object is available to the user, but not currently selected.
- **In Focus:** The object has been activated by a mouse-click, tab key, or script.
- **Hover:** The user pauses the pointer over the object.
- **Pressed:** The mouse button is held down on a clickable object.

Figures 8 and 9 provide an example of using the **Pressed Object State** to provide feedback. In Figure 8, the **Send by Email** button from the Contacts Starter Solution that is included with FileMaker Pro is displaying the **Normal State** styling. When the button is pressed (Figure 9), the button is styled with a different gradient and inner shadow, which gives the impression that the button has been depressed. This subtle change provides feedback that the user has initiated the Send by Email action.

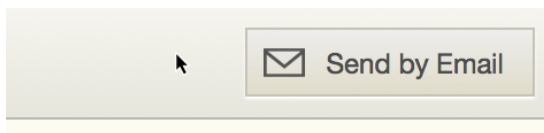


Figure 8

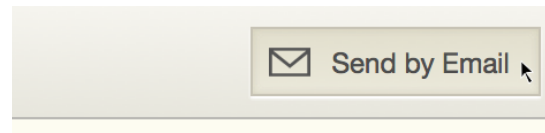


Figure 9

To use the **Object State** styling, you must select an object on the layout, switch to the **Appearance** tab of the **Inspector** (Figure 10 and 11), choose a state from the drop-down menu, and set the style attributes you want the object to have, when it is in its selected state. Figure 10 shows the **Normal State** styling for the Send by Email button. Figure 11 displays the **Pressed State** styling for the same button.

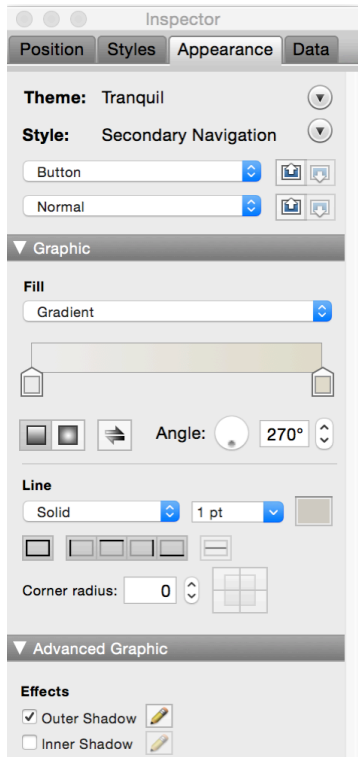


Figure 10

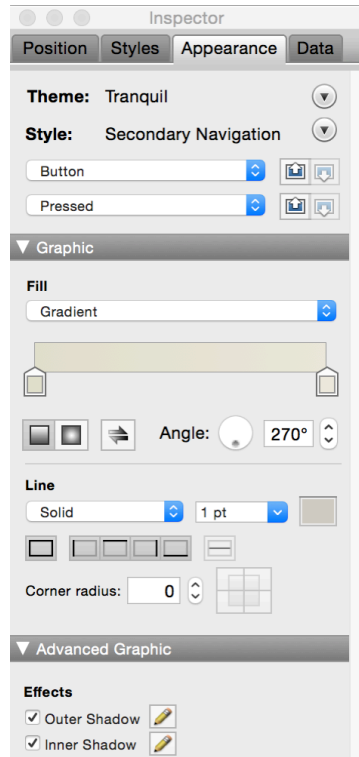


Figure 11

A good example of using visual styling to provide feedback in response to the user's actions can be seen in the Invoice Starter Solution on FileMaker Go on the iPhone. Figure 12 depicts the interface, as it is when the user is viewing an invoice. When the user clicks on the **Edit** button at top right, the color of the header is changed from black to blue, the text of the button is changed from **Edit** to **Done**, a gradient is applied to the **Done** button, and other layout objects are subtly changed.

When the user clicks into date field (thus activating the **In Focus State** styling as depicted in Figure 13), the field is outlined with an outer shadow. When the user clicks the **Done** button, the interface returns to the original state. Each of these visual changes provide feedback to the user about the action they have taken, reinforces their mental model, and decreases the time required to learn the application.

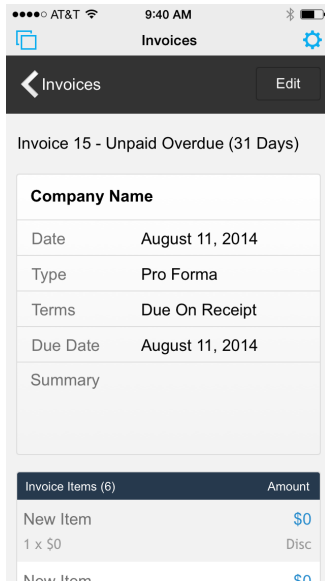


Figure 12

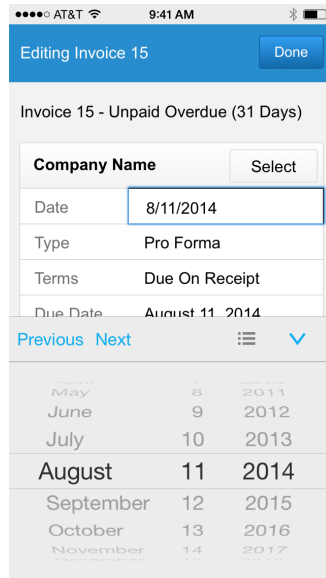


Figure 13

### Provide feedback about scripts and processes that the user must wait to complete

Few things are more frustrating than clicking a button, field, or seemingly interactive element and seeing no response. We wonder whether the computer is doing anything – and we worry that if we click the button again we might break something or create a duplicate entry. The lack of response destroys the user’s trust in your application, and disrupts their attention and flow.

There are a number of ways you can provide users with information during long running scripts and processes. These include updating field data, updating **merge variables**, using the **Set Variable** and **Refresh Object** script steps (as displayed in Figure 14), or creating and displaying a progress bar that gives the user the information they need about the process they have initiated.

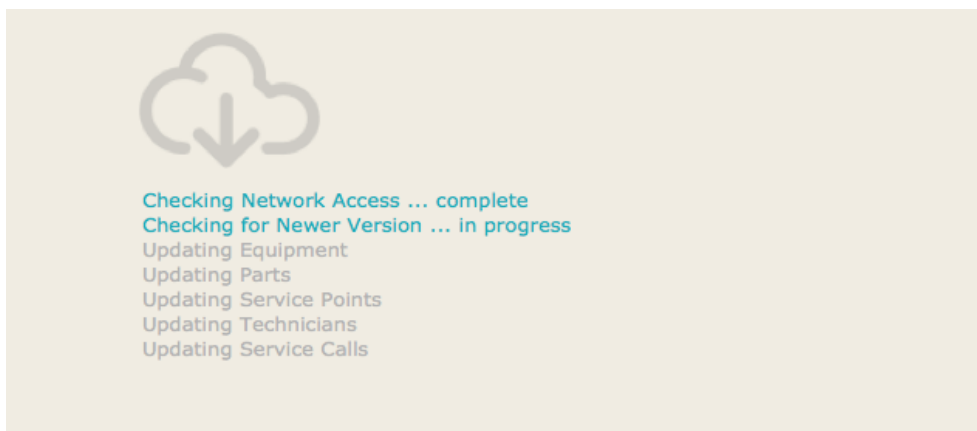


Figure 14

In order to ensure the user sees the information about why they are waiting, place notifications within the area the user is focusing upon. Generally, this is close to the object from which they initiated the action requiring feedback.

For example, when a user presses the **Print Report** button in Figure 15, their attention is focused on the immediate area of the **Print** button. Since they have just pressed the **Print Report** button, users are most likely to see status messages presented in the immediate area of the **Print Report** button; however, if the status message was located in another area of the screen, users could easily miss the message.

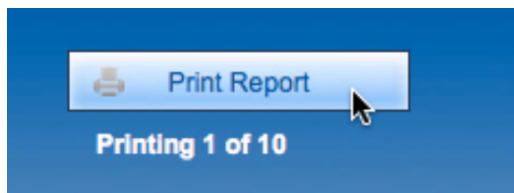


Figure 15

The best way to know if a user is seeing the feedback you are providing is to watch them work. It only takes a few minutes watching a user who is new to an interface, to know whether you are providing enough feedback, and providing feedback within the area where the user is looking.

Read this article for more information about using **Merge Variables**:

[http://www.filemaker.com/help/11/fmp/html/create\\_layout.9.30.html](http://www.filemaker.com/help/11/fmp/html/create_layout.9.30.html)

Read this article to learn more about using the **Refresh Object** step:

[http://www.filemaker.com/help/13/fmp/en/html/scripts\\_ref2.37.62.html](http://www.filemaker.com/help/13/fmp/en/html/scripts_ref2.37.62.html)

### Integrate status information into the interface

Attention is fragile. Protect your user's attention and focus by integrating status information and other feedback into your solution's interfaces, where they can find it within their normal workflow. An example of this is the Assets Starter Solution (Figure 16), which displays information about the availability of the asset in multiple places and forms throughout the interface (textual and visual).



Assigned To	Joe Smith	
Location	Conference Room	
Condition	Good	
Due Date	February 23, 2016	
Days Till Due	365	

Figure 16

## Quick Tips

### Do

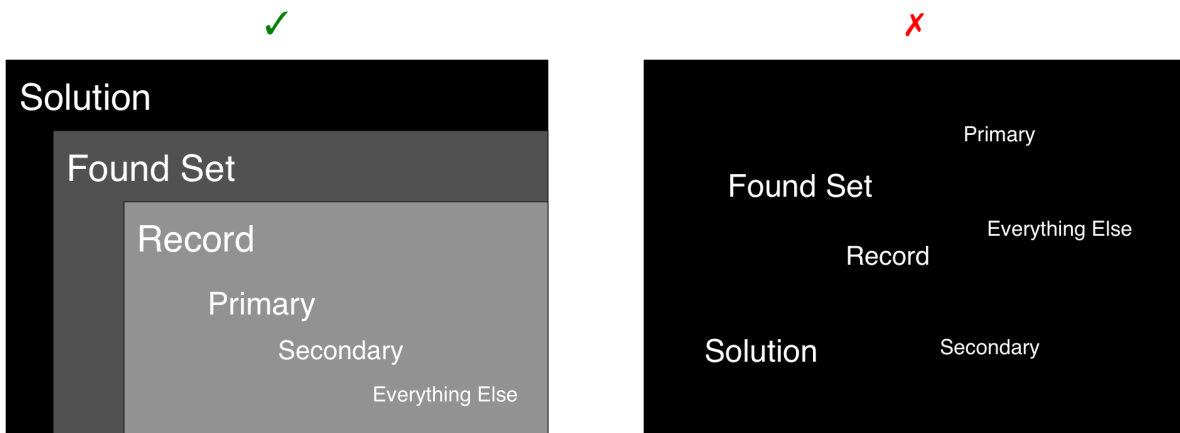
- Ensure that every action a user takes is followed by an understandable and expected reaction.
- Use the **Object-Display-State** styling (**Normal, Hover, Pressed, and Focus**), **Conditional Formatting, Merge Variables**, new windows, and **Custom Dialogs** to provide feedback.
- Use the least obtrusive method possible while giving the user the information they need
- Ensure feedback happens within the time users would expect.
- Ensure feedback happens within the area of the screen where users are focusing.
- Use **Object Display State** styling (**Normal, Hover, Pressed, and Focus**), **Conditional Formatting, Merge Variables**, and **Custom Dialogs** to provide feedback.
- Watch new users work to determine whether your provided feedback is sufficient and effective.

### Don't

- Fail to provide feedback to the user.
- Place critical feedback in a part of the screen that the user may not see.
- Hide status information behind a button or scripted process.
- Interrupt the user with dialogs to provide information that could be integrated into the interface.

## Provide navigation elements to orient and direct your users

*Get the hierarchy right and provide well-labeled landmarks.*



### Define the hierarchy of data and functions

The foundation of most effective navigation systems is a clear hierarchy: sets of groupings and parent-child relationships. These relationships are simple and familiar as we are all constantly interacting with information organized in hierarchies. Examples include: organizational charts, family trees, Amazon's product catalog, and newspapers: the section, page, headline, and subhead organizational.

Well-designed software utilizes a hierarchy that mirrors the way the user would naturally model and organize the concepts represented in the software. Often, the hierarchies in software solutions are organized (depicted above) from the global to the granular:

- data elements and functions to orient the user to and navigate between the major sections of the application
- data elements and functions to orient the user to, navigate between, and perform operations upon records in the found set (Import Records, Export Records, Print Report)

- data elements and functions to orient the user to, navigate between, and perform operations upon the current record (Email Record, Print Record)
- the most important data and functions in the current record
- the second most important data and functions in the current record
- everything else with related elements grouped together

To create a hierarchy, talk to the people who will use your solution about the way they organize the information you are representing in your application. Once you understand the data and functions to be included, categorize each piece of information by answering the following questions:

- Which data and functions are related to or operate upon the solution?
- Which data and functions are related to or operate upon the found set?
- Which data and functions are related to or operate upon the current record?

Then rank the information and functions in the current record in levels of importance and group the related data and functions. As you organize your hierarchy, assess each piece of information to determine where it fits, if it fits, and whether it might belong somewhere other than the current interface.

### Arrange and design layout element styles to mirror the underlying hierarchy

Design interfaces so that users can quickly discern the relationships between elements and the underlying hierarchy. Begin by blocking out the items to be on the interface. Items at the top level of the hierarchy should be visually higher, or have greater weight, than items lower in the hierarchy. Items that are related and are at a similar position in the hierarchy should be visually grouped together spatially within the interface.

An example of this can be seen in the Contacts Details | iPad interface of the Contacts Starter Solution. Once blocked out as in Figure 17, it is apparent that navigation between records occurs at the top left, current record functions at the top right, followed by the Current record Photograph and Summary, and finally groupings of fields related to the current record.

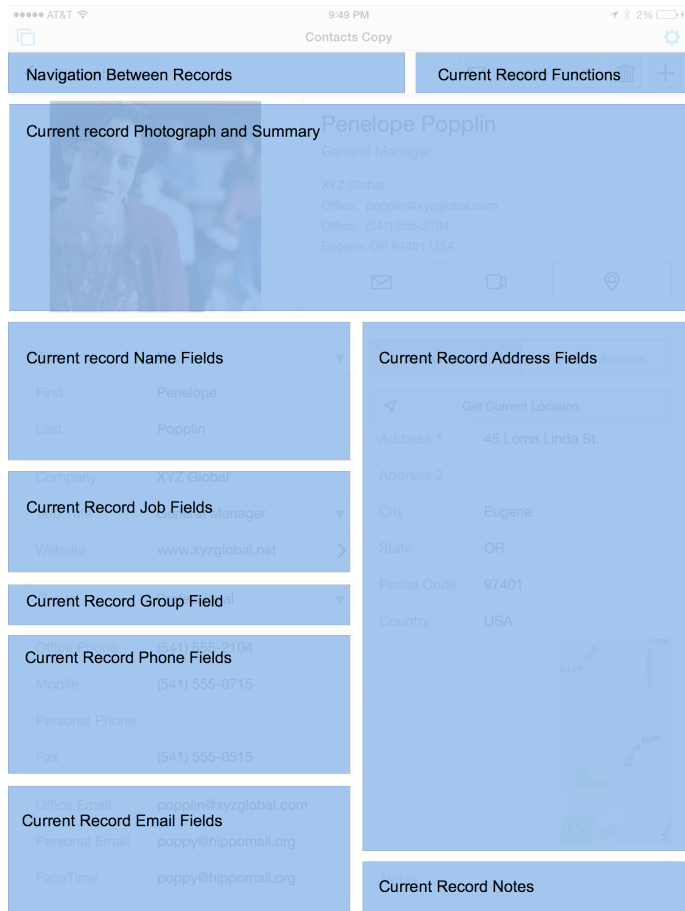


Figure 17

A great way to determine whether your visual presentation reflects your underlying hierarchy is to squint at the interface. With your eyes slightly open, only the most distinct details stand out. If you squint your eyes and nothing pops out, either you have not made clear decisions about the hierarchy of data and functions, or your visual presentation does not communicate your hierarchy.

Consider Figure 18 and Figure 19, which have been blurred to simulate squinting. While you cannot read the text of the navigation tab in Figure 18, the lighter rectangle against the black bar at the top of the interface clearly stands out. The navigation buttons to the right side of the black bar are also distinct. Similarly, the header text (just below the black bar) clearly has more weight than the other text on the layout.

In contrast, in Figure 19, the navigation buttons at the top of the interface are indistinct, as are the action buttons at the top right and the header text. Consequently, the eye has nowhere to land and there is little structure to orient users.





Figure 18



Figure 19

Resist the urge to place elements where there happens to be empty space on your layout. Instead take the time to plan or rework your interfaces, so that you have room to position new elements where they make sense within the overall hierarchy.

#### Provide landmarks to help orient users

Landmarks are navigational elements that help to orient the user to their current location within the larger structure. In a well-designed interface, landmarks function like a “you are here” arrow on a map in a public transportation system. They decrease anxiety and confusion and help the user build their mental model of the solution they are using.

The FileMaker Developer Conference 2014 DevCon2Go solution (Figure 20 and 21) provides several landmarks to orient users within the application hierarchy. Heading each of the interfaces, the Navigation menu allows users to easily switch between the top-level functional areas of the software (i.e., Schedule, Speakers, Maps, Twitter, and Information). On the **Speaker List** page, the list of speakers is grouped alphabetically by last name, which orients users within the list of speakers. On the **Speaker Detail** page, the speakers’ names and photos orient users to the specific speaker they have selected.

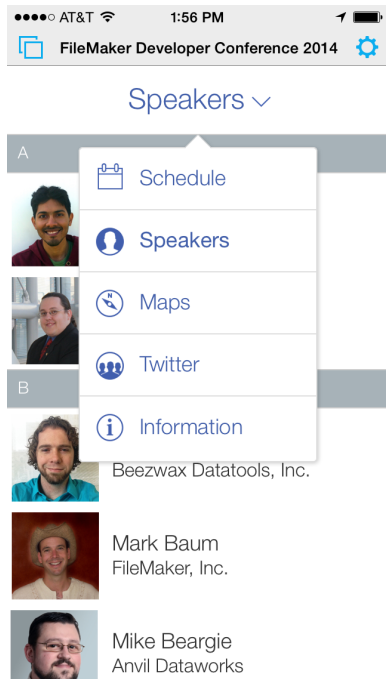


Figure 20

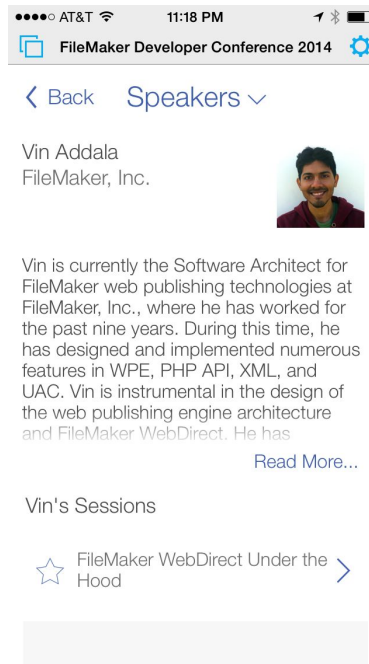


Figure 21

### Clearly label navigational elements

Labels, icons, and other navigational elements facilitate a conversation between you and your users. Every navigational element in your solution should be visible and written in language that the majority of your users can understand. Use as few words as possible, while helping your users differentiate the distinct areas of the software application.

A nice example of this is provided in Figure 22. In this image, the designer from Angel City Data used text to orient the user that they are looking at the record of “Jake Johnson”, large, clear buttons to indicate the three areas of the solution to which the user can navigate (open orders, history, and quotes); and what they are likely to find if they travel the available paths (three open orders, no prior history records, and one quote).



Figure 22

### Quick Tips

#### Do

- Organize your application into a clearly defined hierarchy
- Talk to your users to ensure that the hierarchy represented in your solution mirrors the way they would naturally organize the concepts and functions represented in the software.
- Mirror the underlying hierarchy, with a logical arrangement of elements and visual presentation
- Visually group peer elements
- Squint at your interface, to get a sense of whether the structure is apparent
- Provide landmarks and cues to help orient your users
- Clearly label navigation elements

#### Don't

- Arrange or visually style elements that occur at different places in the hierarchy to look as if they are at the same place in the hierarchy.
- Omit navigation elements.
- Place elements where there happens to be empty space on your layout.

# Make it Efficient to Use

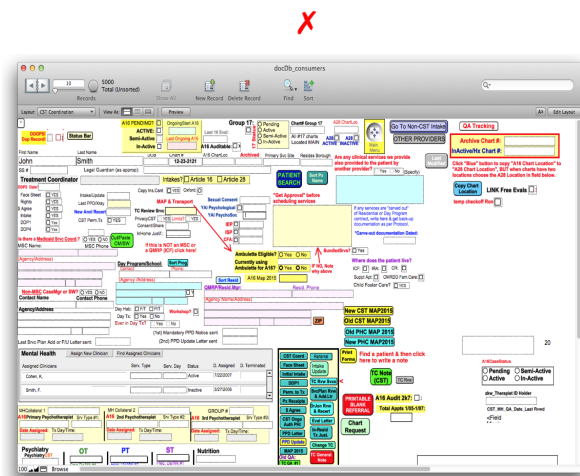
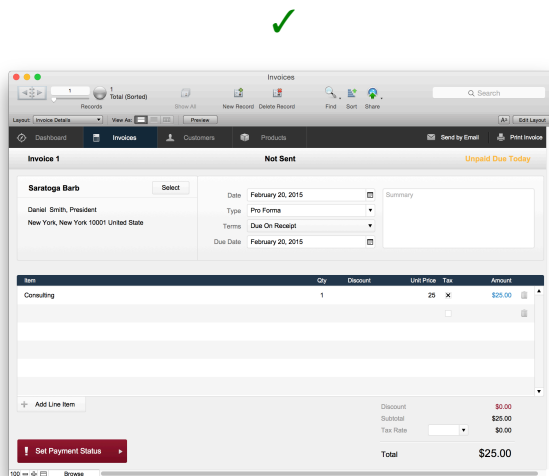
Humans are limited. The amount of information we can process at any point is finite, our attention is fragile, and our working memory is constantly replaced as it is updated with new information. If you want your solution to be efficient for people to use, you have to ensure that the interface and interactions you design minimize the demand placed upon users.

You can reduce the demand imposed by your application, by:

1. Focusing interfaces to help users achieve specific goals.
2. Being consistent and monotonous.
3. Not requiring great precision.
4. Supporting the development of productive habits
5. Anticipating and preventing errors.
6. Not interrupting.

## Focus interfaces to help users achieve specific goals

*Design interfaces for specific scenarios, provide a clear path to completion, and remove extraneous elements.*



## Design for specific scenarios

The best way to make an interface easy to learn and efficient to use is to design each interface to address the goals of people engaged in specific scenarios. As you add new functions to an existing interface, assess the interface to determine if the new functionality enhances the user's ability to achieve the goals for which the interface was designed, or if the new functionality would instead work better someplace else. Be aware that if you are addressing more than a few scenarios in any one interface, you are probably overwhelming your users.

The following steps will help you focus your designs on specific scenarios:

### **Identify the actors**

Begin by identifying all of the people who have some involvement with the problem you are solving. For example, if you are building an invoicing application, the actors might include your customer, the person creating the invoice, the person doing the accounting, and the person managing the business. While the same person may play multiple roles, it can be helpful to think about distinct actors for each different role.

### **Identify the scenarios**

Next identify the scenarios or situations the actors are in when interacting with the problem. For example, in an invoicing solution, the scenarios may include creating an invoice, reviewing old invoices, checking the payment status of an invoice, and receiving payments.

### **Identify the actors' goals in the scenarios**

Identifying the actor's goals one of the most important parts of the software design process. If you correctly identify the users goals, you can design software that solves their problems. If you do not understand their goals, or if you disregard them entirely, you might design a solution that is functional and pretty, but solves the wrong problems.

### **Create a one-sentence problem statement**

Focus your design efforts by creating short-statements that helps you clarify and constrain your solution. The statement should identify the primary scenarios the interface is being designed to address and the actor's goals in those scenarios:

Design a [ form of solution ]  
To enable [ User Type ]  
To accomplish [ Goal ]  
In [ Scenario ]

For example, we used the following problem statement when designing the new invoice flow of the invoices starter solution:

Design a [ workflow for a FileMaker Go iPhone solution ]  
To enable [ a business owner ]  
To [ create an invoice ]  
In [ remote meetings with a customer].

### **Create each interface to enable specific actors to achieve their goals in specific scenarios as captured in your problem statements.**

Design your interfaces with the actors, scenarios, and goals firmly in your mind. Start out by doing a few rough sketches, and only adding as much detail as you have certainty that you have the right solution. As you reach a possible solution, go back to your problem statement and ask yourself whether your solution enables the actor to accomplish the goals in the specified scenarios. If not, go back to the drawing board and keep iterating.

### **Provide a clear path to completion**

Once you understand the scenarios your users face and their goals in each scenario, your objective should be to allow users to complete their work as efficiently as possible. To do so, you first need to give users a clear path through the interface, so they can complete their work.

For example, Figure 23 and Figure 25 present two different layouts that address the same scenario: A clinician in a rehabilitation clinic is documenting her visit with a patient. Both enable the user to record all of the required information; however, Figure 25 provides a much clearer path to completion.

As is visible in Figure 24, the user's attention (and mousing hand) must travel from the top left of the interface, then to the top right, over to the middle left, on to the bottom left, before finally landing on the middle right.

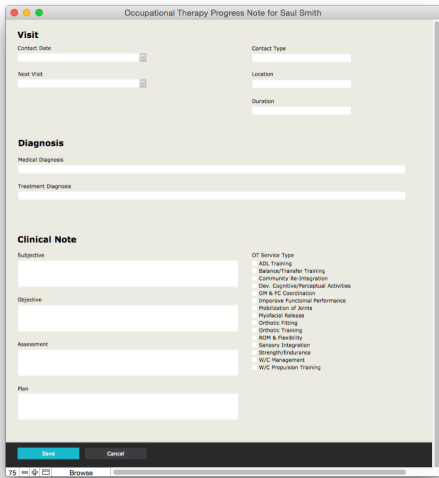


Figure 23



Figure 24

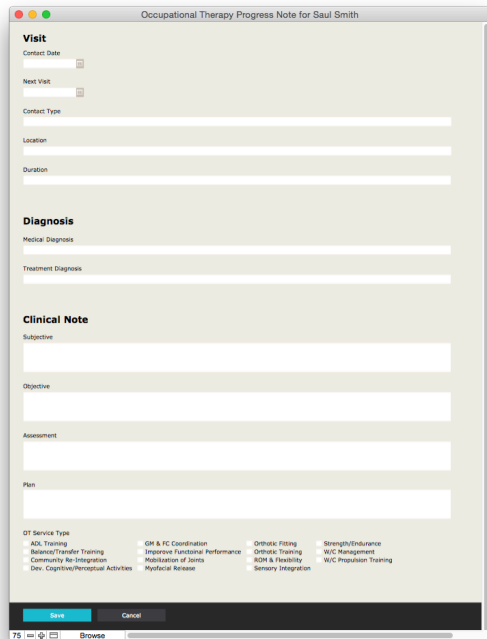


Figure 25



Figure 26

The redesigned layout in Figure 25 has a much clearer path to completion. The eye can scan down the left edge of the interface, which allows us to understand, in one visual scan, all the steps to be completed.

### Delete or move secondary content

Assess each element on the interface, to determine if it helps users achieve their goals. Less important elements may be moved to a pop-over, dialog, or new layout. Delete all extraneous elements.

Hiding secondary content and actions can go a long way toward simplifying your interface. In the **Assets Detail Layout** of the Assets Starter Solution, the interface allows users to add information about an asset, and to check an asset in or out. The information about the asset's check-in or check-out history is important, but it is less important than everything else in the interface. Attempting to jam the history information into the same layout would add noise that detracts from a user's ability to read or edit asset details. By placing the asset history in the popover (Figure 27), the information remains hidden, until users decide to display it.

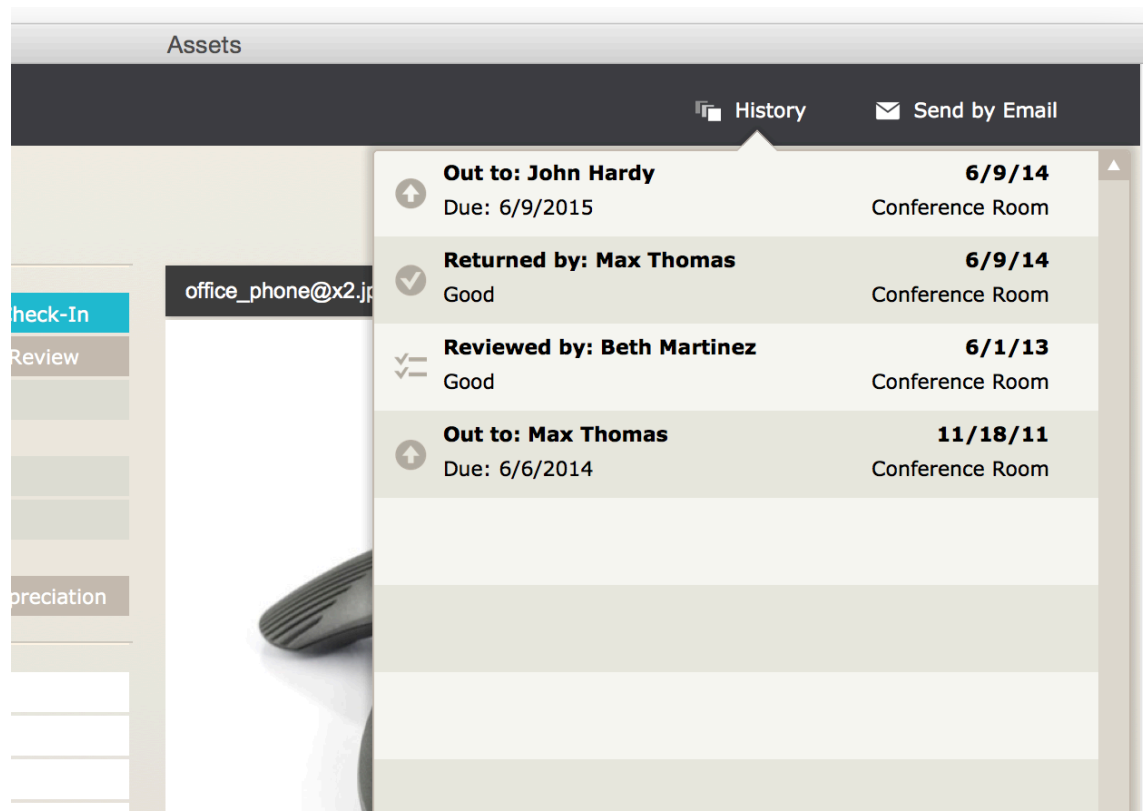


Figure 27

### Quick Tips

Do

- Design the interface specifically to address the goals of specific people (or job types) in a few specific scenarios.
- Assess the interface as you add to it to determine whether the new functionality enhances the existing functionality, or if it would be better placed someplace else.

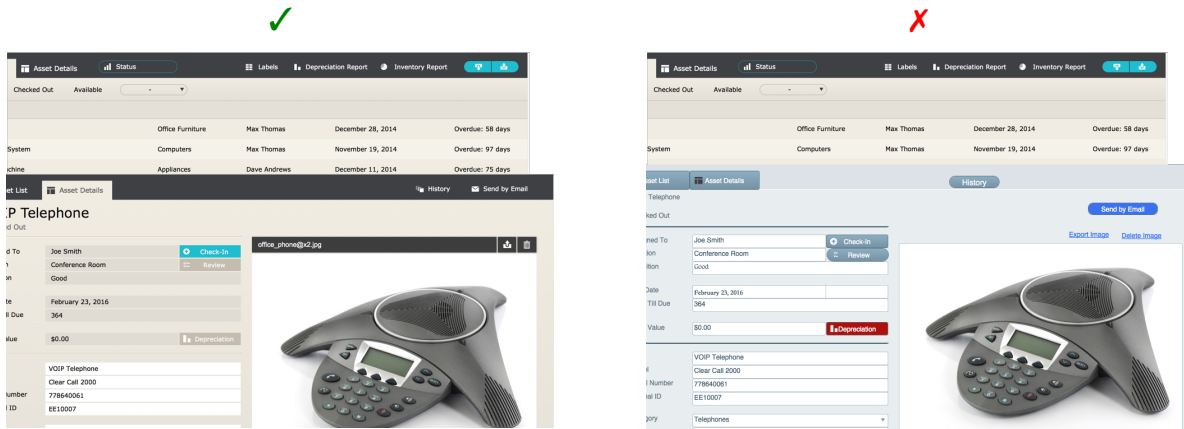
- Provide users with a logical and clear path through the interface.
- Assess each element of the interface, to determine whether it helps users achieve their goals.
- Explore using popovers, dialogs, or new layouts for elements of secondary importance.
- Delete extraneous elements.
- Ensure that your icons are truly iconic or explicitly paired with text.
- Enable **Auto Complete** instead of requiring users to type in a code or ID.

Don't

- Attempt to address more than a few scenarios in any one interface.
- Always include all content and functions related to a scenario in an interface. Instead, consider which can be moved into a popover, window, or secondary layout.

## Be Consistent and Monotonous

*Consistency ensures recognition for learning. Monotony increases efficiency by reducing the number of choices that must be made.*



### Be consistent in the placement, look, and function of elements

An internally consistent interface is one where elements of similar function are styled the same, and function the same, throughout the interface (and application). By being internally consistent you will reinforce the user's mental model and enable them to draw on their previous learning.

When elements that perform the same function appear on two different interfaces which share a similar structure and flow, those elements should be placed in the same position on each interface. Once the positions of elements on an interface are set, they should rarely change.

For example, the "List" interface of the "Assets Starter Solution" (Figure 28) establishes the convention that "New" buttons is found in the top-right of the interface. Because that convention is carried through to the "Detail" interface (Figure 29), regular users of the software will soon develop a habit that allows them to create a new record, simply using their muscle memory of the button's position — without needing to look at the screen.



If instead the designer had switched the position of the **Send Mail** and **New Record** buttons on the **Detail** interface, users would likely find they were repeatedly sending emails, when they really mean to create a new record.

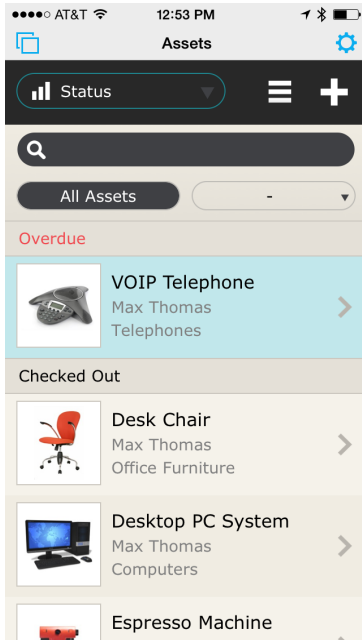


Figure 28

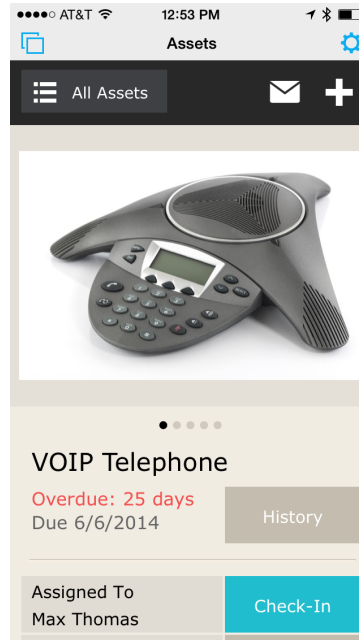


Figure 29

Applying a consistent style to all the elements that function similarly allows your users to apply their existing knowledge to learning new parts of the application. Once they learn that an element is a button, users will expect that all other similarly styled elements will also function as buttons.

For example, the navigation of the Contacts List (Figure 30) and Contact Detail (Figure 31) layouts of the Contacts Starter Solution, the tab elements are the same color and size on both layouts. Similarly, the **Send by Email** button on the **Contact Details** layout is of a similar size, shape and color as the **Labels** and **Print** buttons on the **Contacts** layout.

When you are consistent in styling, positioning, and function of elements, you help users learn new interfaces more quickly, because they can apply the knowledge they have developed while using the software over time.

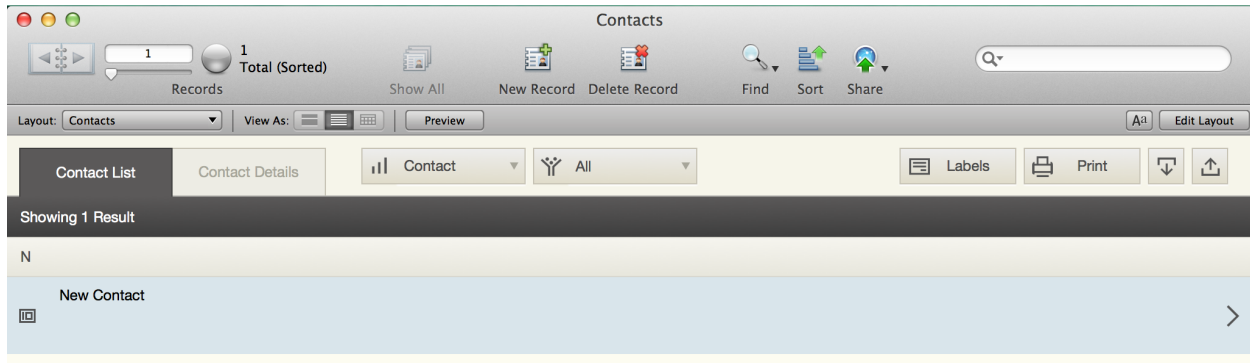


Figure 30

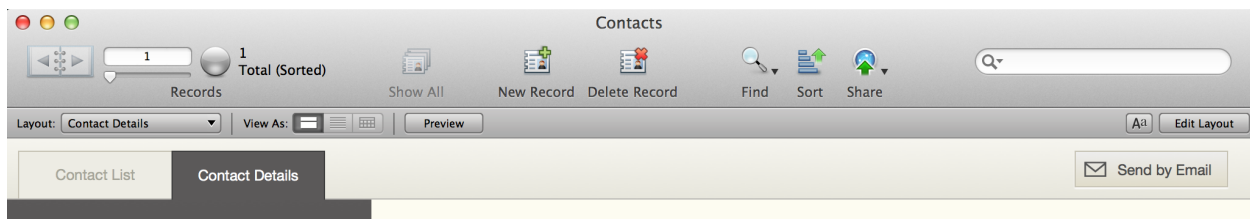


Figure 31

Although it is important to be internally consistent within and between the interfaces across a single platform, it is important not to just blindly apply the same styling and positioning to interfaces on another device platform (e.g., from a desktop application to a mobile app). Doing so risks creating interfaces that are perfect on one platform, but not quite right on another. It is more important to ensure that users can leverage existing knowledge of the application on the initial platform, in order to intuitively recognize the controls on the new platform.

### Be monotonous

A monotonous interface is one in which there is only one-way to perform each kind of function. By designing monotonous interfaces you will increase the user's efficiency and reduce the number of choices the user must make.

Resist the temptation to add more than one button for a single function on any interface. Users are most efficient when there is only one way to achieve any desired result. When there is more than one button for a single function, users must choose between the multiple ways of invoking the function. The time required to make this decision increases proportionally with the number of available options.

For example, on the **Invoice Detail** layout of the **Invoices Starter Solution** and —and indeed throughout the entire solution — there is only one way to email or print an invoice: select the invoice (which switches you to the detail layout) and then scroll down to reveal the **Print** and **Email** buttons (Figure 32).

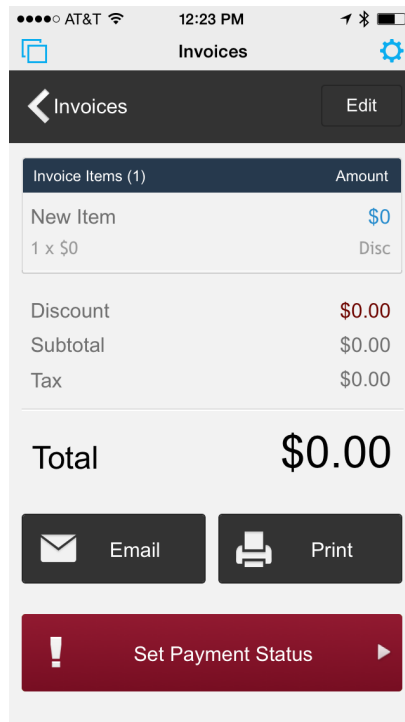


Figure 32

### Quick Tips

#### Do

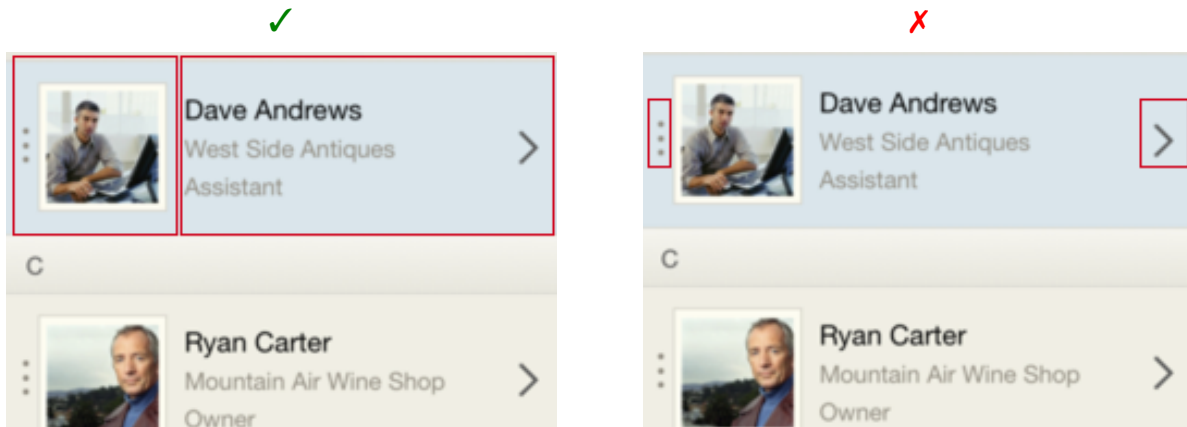
- Apply a similar visual style to elements that perform similar functions.
- Place elements that perform the same function on interfaces that share a similar structure and flow in the same place on both interfaces.
- Resist the temptation to add more than one button for a single function on any interface.

#### Don't

- Frequently move elements on interfaces to which users have become accustomed.
- Blindly apply the same styling and positioning to interfaces on another device platform (e.g., from a desktop application to a mobile app). Instead, strive to create a sense of familiarity which will enable users to leverage existing knowledge of the application on the initial platform.

## Don't require great precision

Maximize target sizes and minimize the distance to target.



### Don't size targets smaller than the minimum recommended size

Clickable targets on desktop interfaces should be at least 12 x 12 points. Frequently used buttons on a desktop interface should be at least 24 x 24 points. Touch targets on any touch interface should be at least 44 x 44 points.

As you can see in Figure 33, only the top of the pointer icon fits in a target sized 12 x 12 points and a standard size finger can only acquire a target on a mobile device that is at least 44 x 44 points. Buttons or targets smaller than these recommended sizes will be very difficult for most people to click or tap. Targets smaller than those recommended will increase the precision required to acquire a target – and will likely increase the frequency with which a user makes errors.



Figure 33

### Maximize target size

The larger the size of the buttons or areas on the screen users try to press, enter, or click, the less precision they will need to click them. You can take advantage of spacing around buttons and other click targets to increase the target size and decrease the possibility that users will hit something else by mistake.

For example, the Contact List Interface of the Contacts Starter Solution on the iPhone (Figures 34 and 35) contains two icons: in the far left of each row, there is a vertical icon composed of three dots; at the far right of each row is a right-facing caret. Both icons are smaller than the 44 pixels recommended for touch-screen targets. If the buttons underlying those icons were sized the same as the icons (Figure 35), the targets would be too small for most people to accurately acquire.

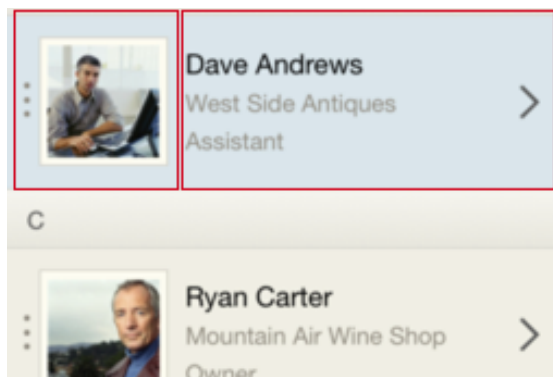


Figure 34

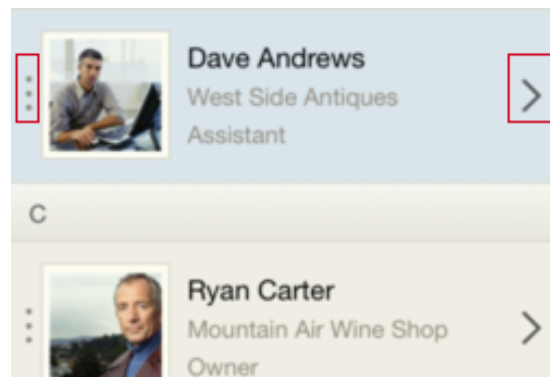


Figure 35

### Minimize the distance to the target

The physical demand an interface places upon a user increases with the distance the user has to move their finger or mouse to click each button, scroll bar, field, or element they need to click or touch. Objects that are closer to a user's starting point, which cover a larger area, require less movement and are less demanding. Objects that are smaller, or those placed farther away are more demanding.

Minimize the distance the user must travel to acquire a button or field, by designing the interface with a clear hierarchy that matches the user's mental model and an efficient path to completion.

### Quick Tips

#### Do

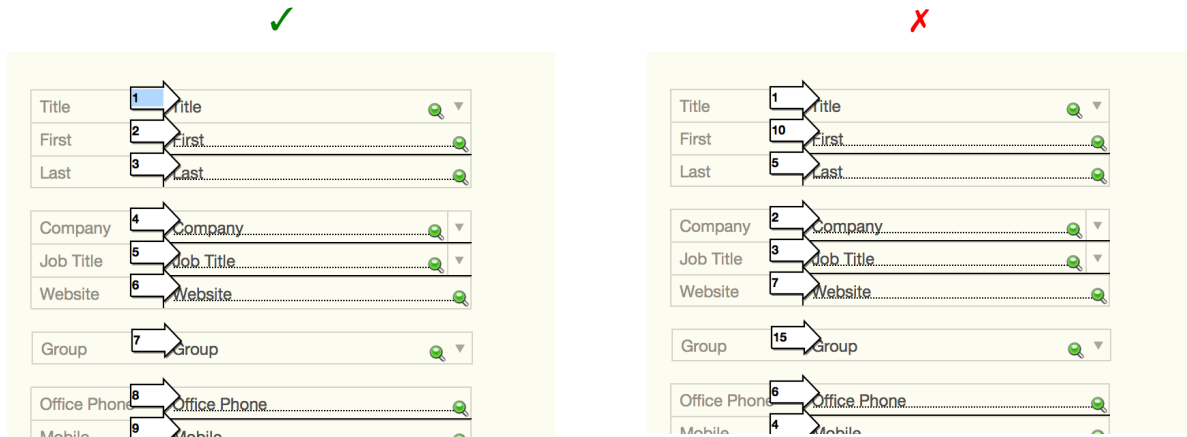
- Size desktop click targets at least 12 x 12 points.
- Size touch-screen click targets at least 44 x 44 points.
- Take advantage of spacing around buttons and other click targets to increase size.

#### Don't

- Size click or tap targets smaller than the recommendations.

# Support the Development of Productive Habits

Encourage habits that allow people to work faster



## Set a logical tab order

To provide a clear path to completion, the tab order should be set to ensure movement through the interface (via tabbing or exiting a field) occurs in the order users would logically expect. FileMaker Pro's default behavior is to move from the top left to the bottom right of the screen. Once you have finalized ordering fields and buttons on your layout, make sure that the tab order is still set correctly.

To set the tab order, enter **Layout** mode and then choose **Set Tab Order** from the menu. For more information on how to set the tab order, refer to this FileMaker Help article:

[http://www.filemaker.com/help/13/fmp/en/html/edit\\_layout.10.56.html](http://www.filemaker.com/help/13/fmp/en/html/edit_layout.10.56.html)

## Set field exit options

Clicking **Tab**, **Return**, and **Enter** keys are among users' most deeply ingrained habits, across all device platforms. Set field exit options that will allow users to navigate to the next field without thinking, and without having to physically move or point to the next field.

To set the keys that may be used to exit a field select the field, navigate to the **Behavior** area of the **Data** tab of the **Inspector** (Figure 36), and check which of the three field exit options (Tab, Return, and/or Enter) you want to result in going to the next field.

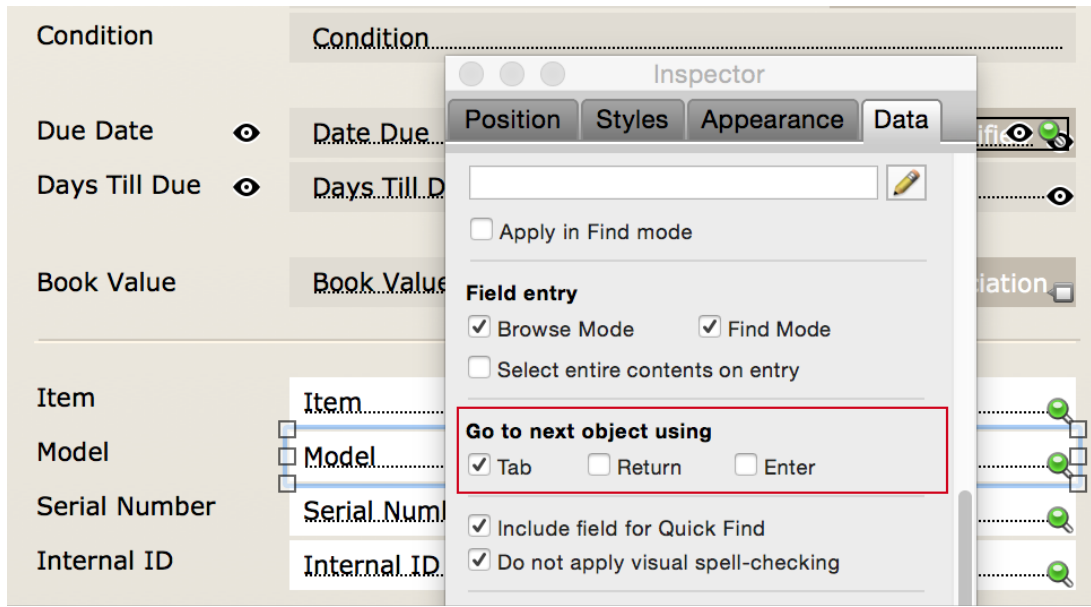


Figure 36

For more information on how to set field exit options, refer to the FileMaker Help article here: [http://www.filemaker.com/help/13/fmp/en/html/edit\\_layout.10.59.html](http://www.filemaker.com/help/13/fmp/en/html/edit_layout.10.59.html)

### Enable the use of keyboard shortcuts

FileMaker Pro users will benefit from keyboard shortcuts that allow them to evoke actions without mousing or clicking. You can enable keyboard shortcuts by enabling the visibility on the most frequently used scripts (Figure 37), creating **Custom Menu Items**, or enabling the **OnLayoutKeystroke** script trigger.

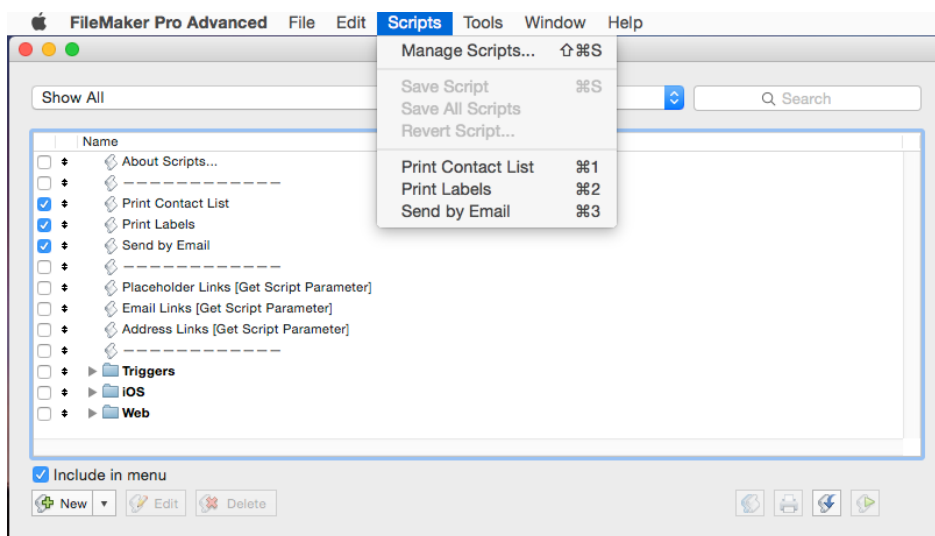


Figure 37

Be careful not to override a FileMaker application shortcut or system shortcut, as users might have already developed habits involving that particular key-stroke combination.

For more information on creating custom menus, read this FileMaker Knowledge Base article:  
[http://www.filemaker.com/help/13/fmp/en/html/fmpa\\_custom.23.8.html](http://www.filemaker.com/help/13/fmp/en/html/fmpa_custom.23.8.html)

For more information on using the **OnLayoutKeystroke** script trigger, see this FileMaker Knowledge Base article:  
[http://www.filemaker.com/help/13/fmp/en/html/script\\_trigg.38.12.html](http://www.filemaker.com/help/13/fmp/en/html/script_trigg.38.12.html)

### Be consistent in how you configure custom dialogs

When defining a **Custom Dialog**, be consistent about the actions you assign to the buttons. Users will become accustomed to frequently used dialogs and stop reading them. To ensure that users do not inadvertently select the opposite of the action they intend, be consistent in the assignment of the default and secondary buttons.

In general, destructive actions should never be assigned to the default button, because users will often habitually click the default button, without reading the dialog's message. Instead, assign destructive actions to buttons two or three (as depicted in Figure 38).

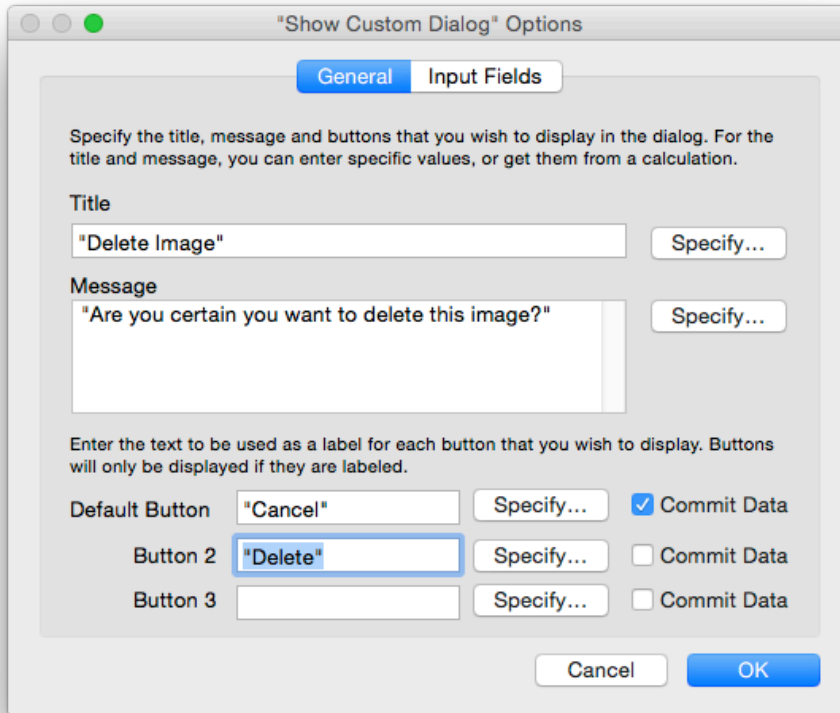


Figure 38

For more information on configuring the buttons of a custom dialog, see this FileMaker Knowledge Base article:  
[http://www.filemaker.com/help/13/fmp/en/html/scripts\\_ref2.37.60.html](http://www.filemaker.com/help/13/fmp/en/html/scripts_ref2.37.60.html)



## Quick Tips

### Do

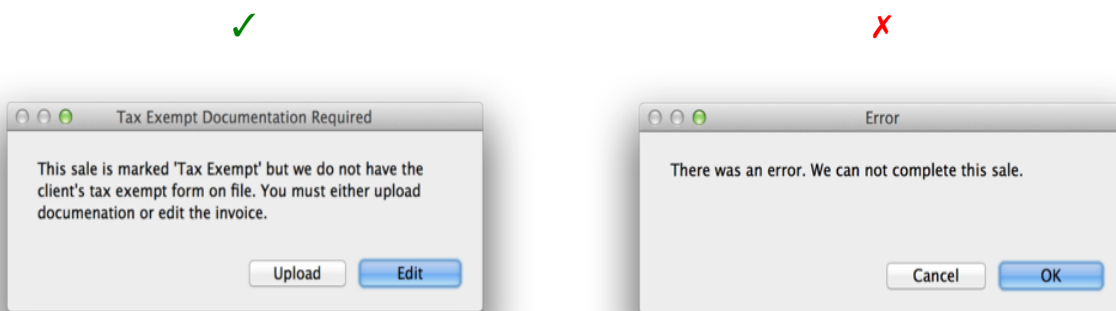
- Set a logical tab order to ensure movement through the interface occurs in the order users would expect.
- Set the Field Exit Options to allow the user to quickly navigate to the next field.
- Enable the user of keyboard shortcuts.
- Be consistent in assignment of buttons in custom dialogs.

### Don't

- Assign the default button in a custom dialog to a destructive action.

## Anticipate, Prevent, and Help Users Recover from Errors

*Prepare your solution to detect errors and enable users to recover from mistakes.*



### Identify potential errors

There are two types of errors you need to account for when designing an interface: data-validation errors and usability errors.

You can identify validation errors by finding all fields that have one of the **Require** options enabled on the **Validation Tab** of the **Field Options** dialog (outlined in red in Figure 39).

The best way to identify usability issues errors is by asking people who have not been involved in the programming to attempt to complete the scenario for which you are designing the interface. Give them the scenario; ask them to think out loud as they attempt to complete the scenario; and then sit quietly and observe the parts of the interface that cause the most problems.

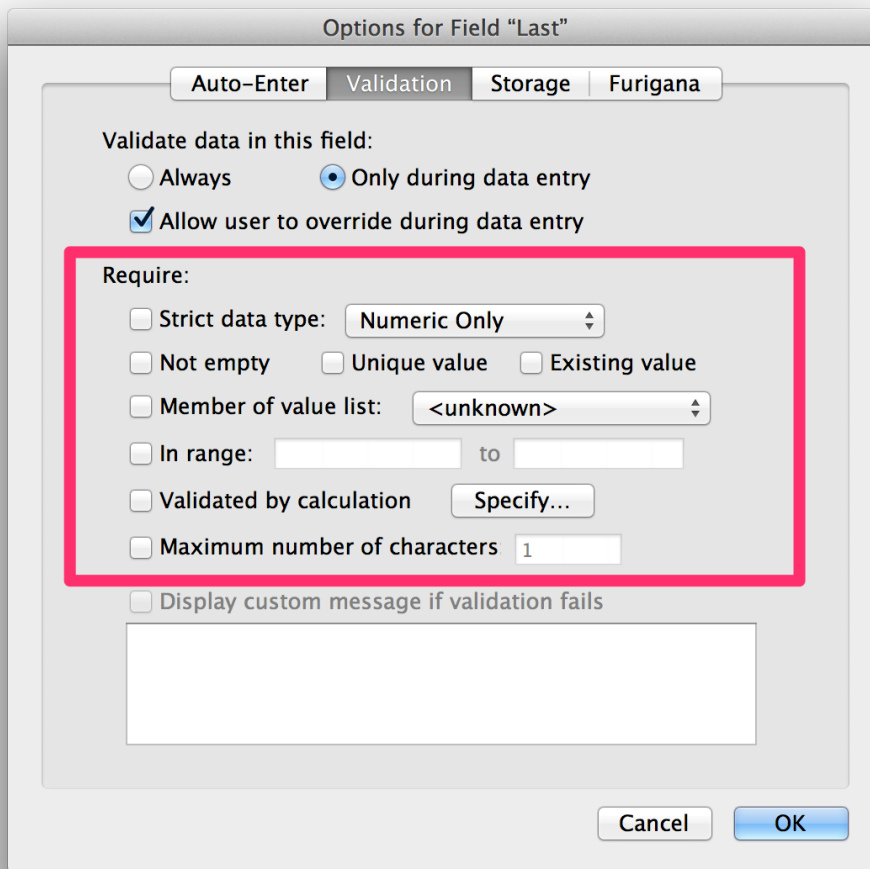


Figure 39

### Highlight required fields

One common error occurs when users fail to enter required data. These errors are preventable by visually differentiating the fields that require data entry from those that do not.

If a layout contains a lot of input fields, but only a few of them are required fields, use text or field formatting to visually indicate the fields that are required (e.g., add the word "Required" next to the field label).

### Constrain choices

Fields that required specific data (e.g., user ID, or the name of an organization or department) are another frequent source of data-entry errors. You can prevent these errors, by constraining the choices available to users.

FileMaker Pro offers several control types designed to help constrain user data-entry choices. These controls include drop-down lists, pop-up menus, check-box sets, radio-button sets, and drop-down calendars. You can also build pickers windows and popovers that provide a filtered list or portal for selection.

- Use **radio buttons** when you want to allow users to choose only one of a few listed options. (Though FileMaker allows it, you should never use a radio button when you want users to choose more than one option).
- Use **check boxes** when you want users to choose one or more of the listed options.
- Use **drop-down lists** when you want users to choose an item from a list of items, but want to provide them with the ability to type in a choice, as well.
- Use **pop-up menus** when you want users to choose one item from a longer list of items than would be appropriate for smaller radio buttons. (Though FileMaker allows it, you should never use a pop-up menu when you want users to choose more than one option).
- Use a **drop-down calendar**, when you want users to enter a date.
- Use a **filtered picker window or popover** (created by placing a filtered portal in a window or popover), when you need to provide users with the ability to select from a large set of items.

### Provide inline validation

Validation errors can be quickly resolved by providing inline validation. Inline validation indicates whether a valid or invalid entry was given, and provides suggestions about how to make a valid entry.

You can provide inline validation, using a combination of object hiding, script triggers, merge variables, and the **Refresh Object** script step. For example, in Figure 40, an **OnObjectKeystroke** script trigger is being combined with a **Merge variable** and **Refresh Object** script step to provide the user inline validation about the number of character remaining. In Figure 41, an **OnObjectExit** is being combined with a text label and object visibility to provide the user inline validation about the expected dates.

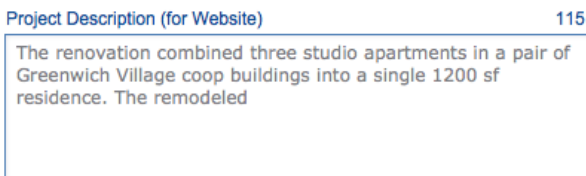


Figure 40

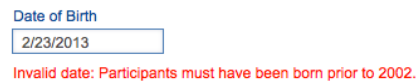


Figure 41

### Provide actionable error messages

Error messages can be frustrating and annoying, so you should only show error messages when there is a problem that must be corrected before users do anything else, or when your application has encountered an unrecoverable error.

Useful error messages help users uniquely identify the source of the problem and recover gracefully. They should be written as a brief readable sentence or phrase that is in keeping with the tone of the software application.

The message depicted in Figure 42 neither helps users identify the source of the problem, nor provides them with a clear action they should follow to recover from the problem. In contrast, the message in the Figure 43 helps users identify the exact cause of the problem and shows the steps they can take to resolve that problem.

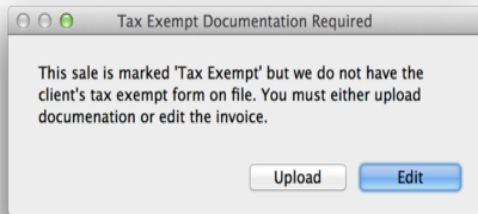


Figure 42

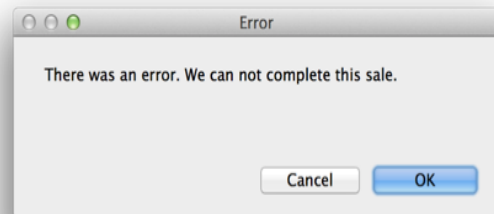


Figure 43

### Quick Tips

#### Do

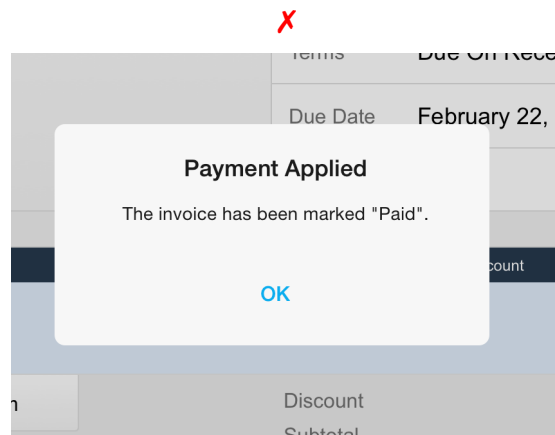
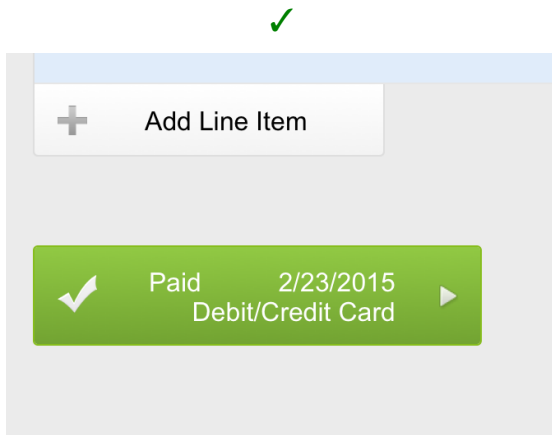
- Identify potential data validation errors by making a list of fields that have one of the **Require** options enabled on the **Validation Tab** of the **Field Options** dialog.
- Identify usability issues by watching users attempt to complete the scenario for which you are designing the interface.
- Highlight required fields.
- Use radio buttons, check boxes, popup menus, drop down lists, drop down calendars, and filtered picker patterns to constrain choices available to users.
- Provide inline validation to indicate whether a valid or invalid entry was given, and provide suggestions about how to make a valid entry.
- Write error messages that help users uniquely identify the source of the problem and recover gracefully.

#### Don't

- Use radio buttons when you want to allow users to choose more than one option.
- Use popup menus when you want your user to allow users to choose more than one option.
- Assign the default button of custom dialogs to destructive actions.

## Don't interrupt

Respect your user's attention, and use custom dialogs and modal windows sparingly.



### Modal windows

A modal window is a child window that opens from an interface that freezes the parent window or application in order to focus the user's attention and action. Modal windows interrupt users, and force them to shift their focus from what they were doing to the window. Consequently, they should be used sparingly.

Modal windows are created in FileMaker Pro when you specify the **Dialog Window (Modal)** style in the **Specify Advanced Style** options of the **New Window** script step (Figure 44) or when you add a **Pause Script** step with an indefinite duration to a script (Figure 45).

Modal windows should be reserved for times when you absolutely cannot continue the flow, without the information being collected in the modal window.

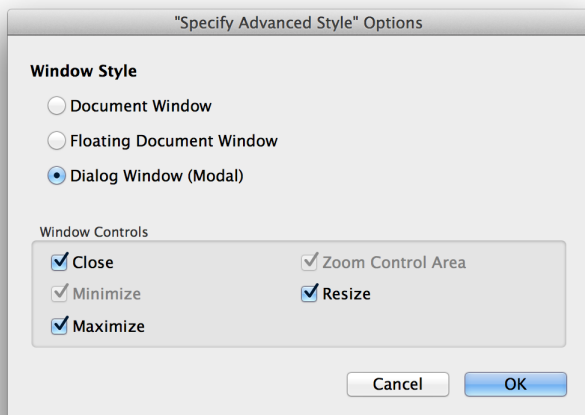


Figure 44

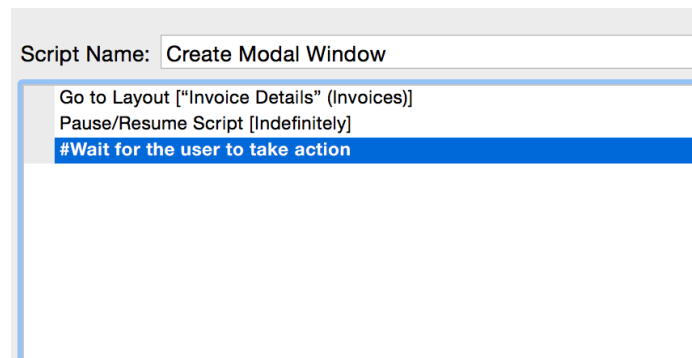


Figure 45

## Custom dialogs

Custom dialogs also interrupt users — forcing them to shift focus from what they were doing to the dialog. When there are too many interruptions, users start to ignore them. Before using a custom dialog, consider whether there's any other way to design the system to eliminate the interruption.

Custom dialogs are best used to alert users to information that could impact their use of the software or their data; or to receive confirmation before initiating a long running or dangerous task, like deleting records. Custom dialogs are generally not needed to provide users with information about the software's standard functions, to update them about normally progressing tasks, or to ask them for confirmation of user-initiated tasks. Instead, as much as possible design your solution so that user initiated tasks can be reversed and integrate status information into the interface.

### Quick Tips

#### Do

- Only use modal windows when you absolutely cannot continue the flow, without the information being collected in the modal window.
- Use custom dialogs to alert users to information that could impact their use of the software or their data; or to receive confirmation before initiating a long running or dangerous task.
- Avoid the need for custom dialogs by designing your solution so that user initiated tasks can be reversed.

#### Don't

- Interrupt – unless it is absolutely necessary.
- Use custom dialogs to present users with information about the software's standard functions, to update them about normally progressing tasks, or to ask them for confirmation of user-initiated tasks.

## Credits

Dr. Don Levan is an experience designer and Design Process Consultant who helps business owners design and build the operations software and mobile apps they need to grow their businesses. A recognized FileMaker expert, Dr. Levan was hired by Apple Inc., to provide consulting to FileMaker on the design of the company's Starter Solutions and product features.

In addition to being a frequent speaker at the FileMaker Developers Conference, PauseOnError conferences, and other software industry events, Dr. Levan provides design consulting and training to organizations and teams.

## Legal Notices

©2015 FileMaker, Inc. All rights reserved. FileMaker is a trademark of FileMaker, Inc., registered in the U.S. and other countries. The file folder logo is a trademark of FileMaker, Inc. All other trademarks are the property of their respective owners. Product specifications and availability subject to change without notice.

THIS DOCUMENT IS PROVIDED «AS IS» WITHOUT WARRANTY OF ANY KIND, AND FILEMAKER, INC., DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THE WARRANTY OF NON-INFRINGEMENT. IN NO EVENT SHALL FILEMAKER, INC., OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, LOSS OF BUSINESS PROFITS, PUNITIVE OR SPECIAL DAMAGES, EVEN IF FILEMAKER, INC., OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY. FILEMAKER MAY MAKE CHANGES TO THIS DOCUMENT AT ANY TIME WITHOUT NOTICE. THIS DOCUMENT MAY BE OUT OF DATE AND FILEMAKER MAKES NO COMMITMENT TO UPDATE THIS INFORMATION.