



When Auto-Save Is Not An Option

How to design for a “Save” button in next-gen enterprise applications

JULY 2024

BY

SHON DeCAMP

SR. PRODUCT/USER EXPERIENCE DESIGNER

Introduction & Background

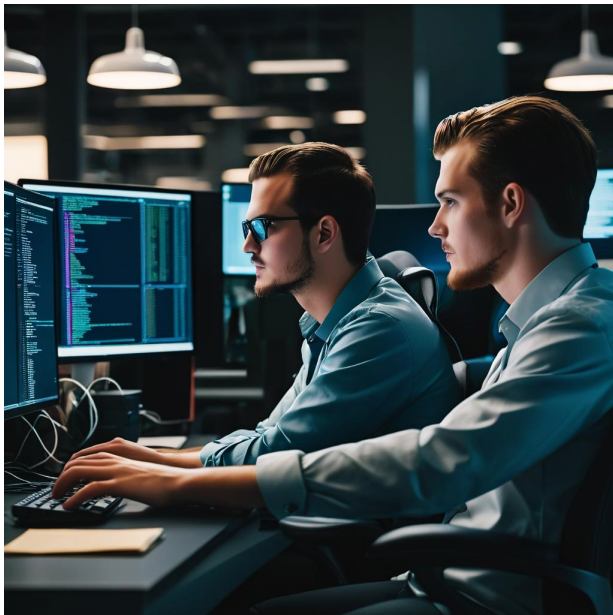
The hard truth is, some databases and APIs from 20 years ago aren't going to change before your next project. The best possible experience might not include all modern best practices. Designers can add significant value in making a 2005 product look and feel like a 2025 product.

A frequent issue is the inability to support auto-save. Sometimes the schema or APIs won't support it. Sometimes the front-end architecture isn't built for it. Whatever the reason, sometimes getting autosave in your project won't be feasible. So, what to do?

This was my situation when I began re-designing a legacy application at Accela. A new front-end was needed, but there weren't resources for new API work or Database work.

First, some general advice

Get familiar with the UI. Work closely with developers to understand the front-end architecture, how the code is structured, and why a save button is required.



My first step was to set up a series of training sessions with the lead front-end developer.

For 30 minutes every week we would take an informal look at how the UI came together.

These meetings were recorded and documented, then used later for designers joining the team.

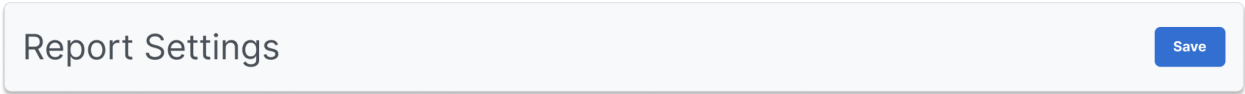
Find a collaboration method that works well for your cross-functional team.



Guidelines to Use

Keep the “Save” button

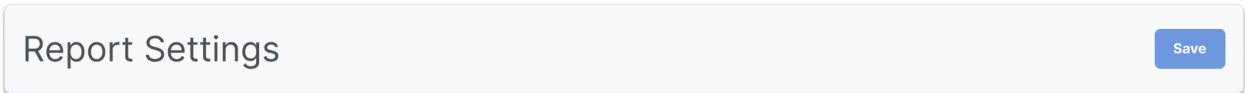
Unless the front-end team has a sustainable, repeatable, reliable method to “fake” auto-save, then keep the “Save” button in the UI. It may likely be the main Call To Action - representing the completion of a task.



Reasons not to get rid of the button:

- Triggering the Save function with other page interactions is trickier than it seems
 - Every navigation option (including closing the browser tab) would have to be wired to trigger the save function
 - No action at all would also have to trigger the save function (i.e. the user walked away thinking everything was saved)
 - The save function on a legacy product will not be as quick. Older functions are designed to send everything in the UI to the database, not just the changed information. Save functions can easily pile up to the point that the browser can’t handle them all. Data can be lost.
- Save buttons don’t bother users who are moving from a 1995 experience to a well-executed 2025 UI. In fact, a lot of users making this transition enjoy the physical confirmation of clicking a button.

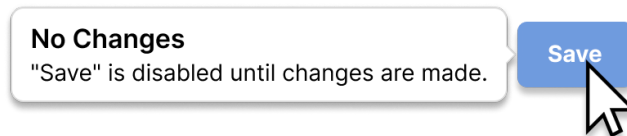
Avoid disabled buttons



- Adding the “disabled” attribute to an HTML button makes it difficult for screen readers to pick up or read properly. Of course there is a work-around for

everything, but unconventional UI is the opposite of the goal in most enterprise operations.

- Disabled buttons place unnecessary cognitive load on users. This can be solved by one of two undesirable approaches:
 - Tooltip or other messages - this approach still requires the user to interact with or at least find the disabled button to discover what is wrong with their form entries.



- Constant error state validation - starting a form with a bunch of red messages is about as off-putting as a designer can make an already cumbersome task.

A screenshot of a web application interface for "Report Settings". At the top, there's a breadcrumb "Standard Choices > Adhoc_Report_Settings". The form has a "Save" button in the top right. A prominent red error bar at the top says "Error One or more fields are invalid." Below this, there's a toggle for "Enabled" which is turned on. There are radio buttons for "System Switch", "Shared Dropdown", "EMSE", and "Business Configuration" (which is selected). Below the radio buttons is a dropdown menu labeled "Select a Business Configuration Option" with a red error message "A Business Configuration Option is required". There's also a text input field labeled "Enter a description". At the bottom, there's a table with columns: "Value", "English (AU) Value", "Description", "Status", and "Actions". The table has one row with the value "ACA_Configs_Tabs". A "+ New" button is in the top right of the table area. A search bar and a language selector "English (US) - default" are also present.

- Users thrive in a UI that makes them feel smart. Seeing an opaque CTA (with or without a clarifying message) does the opposite. It creates questions, doubt, and discouragement.

Don't change the button

It is natural to start exploring ways to use button properties to communicate whether changes have been saved. Here's why that might not work:

- Icons can be unclear, harder to remember, and difficult to customize. The cognitive load increases when the button icon changes.



- Changing among colors (success, primary, danger, etc.) or hierarchy (primary, secondary, tertiary) is generally a bad practice for most components, especially buttons.



- Text is an ideal way to achieve explicit communication with users. But the text in a button label is the wrong place to do it. A button label should be constant. One exception may be if a spinner icon is animated while the save function is being performed. In this case, the label could change from "Save" to "Saving" temporarily.



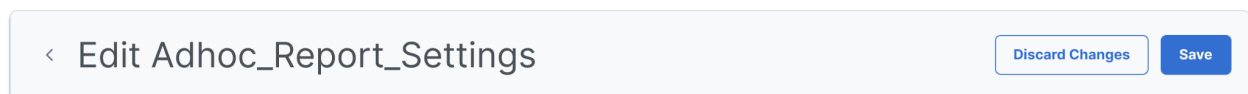
"Discard Changes"

While this button should be completely avoided in consumer experiences, it can have a place in enterprise applications, especially when transitioning to a modern design.

Consider a "Discard Changes" button when...

- The user has deliberately entered a "side quest". This might happen when the user navigates from a read-only version of a page to an editable version of a page.
- The user is interacting with a large and/or complex form where starting over is easier than making edits.

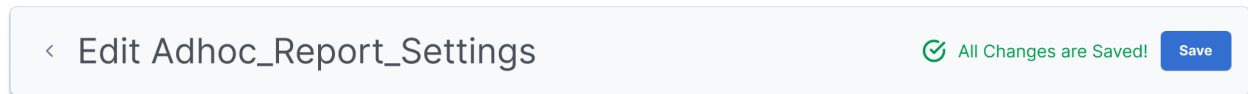
"Discard Changes" should always be near the "Save" button, but not share the same hierarchy as the "Save" button.



Proper affordance to communicate if changes are saved

Don't make the user work for the info.

- Avoid requiring the user to hover, click, or scroll to confirm whether changes are saved.
- Use text to be unambiguous
- Keep it in the same location in the view, so the user can develop muscle memory
- Keep it near the “Save” button to allow for a convenient response



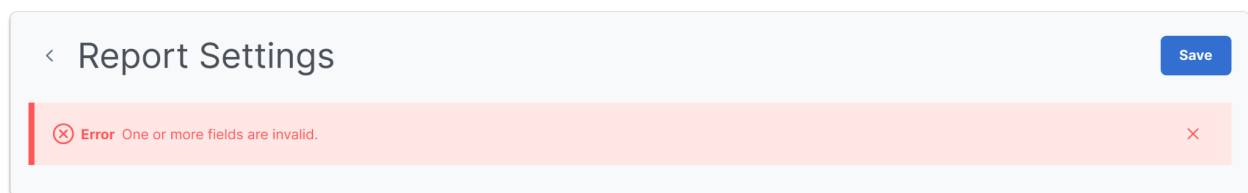
Dialogs and Alerts

- If the user does accidentally attempt to perform an action that would cause desired changes to be lost, then provide an interruption.
- Accept the fact that a destructive action may be performed despite the best design

Validation and Errors

Clicking the “Save” button is likely the first opportunity for specific/system validation to run, especially in applications using older APIs. That means there are important considerations.

- Use a parent message as a blanket notification of errors
 - Keep it near the save button
 - Endure that it's position is fixed in the view
 - Don't allow it to change the scroll position
 - Make sure it gives the user clear affordance for locating, understanding, and correcting errors.
 - Put the user in charge of its visibility



- Ensure that proper error states and error messages are available for individual inputs

Standard Choices > Adhoc_Report_Settings

<

Edit Report Settings

All Changes are Saved!

Save

Enabled

System Switch

Shared Dropdown

EMSE

Business Configuration

Select a Business Configuration Option

Enter a description

Values

+ New

Q

English (US) - default

Value	English (AU) Value	Description	Status	Actions
ACA_Configs_Tabs	ACA_Configs_Tabs	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sec	Enabled	
ACA_Configs_Tabs	ACA_Configs_Tabs	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sec	Disabled	

Standard Choices > Adhoc_Report_Settings

<

Edit Report Settings

Discard Changes

Save

Error One or more fields are invalid.

Enabled

System Switch

Shared Dropdown

EMSE

Business Configuration

Select a Business Configuration Option

A Business Configuration Option is required

Enter a description

Values

+ New

Q

English (US) - default

Value	English (AU) Value	Description	Status	Actions
ACA_Configs_Tabs	ACA_Configs_Tabs	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sec	Enabled	