



## Efficient Useful GUI design Patterns using [ezAJAX™](http://www.ez-ajax.com)



Notice: This document and the conceptual designs contained herein may not be printed, copied, reproduced or distributed without the expressed written permission of Hierarchical Applications Limited, Inc. All Rights Reserved.



Preface .....	3
Problem Statement .....	3
Why AJAX ? .....	3
Revised GUI Sample using AJAX .....	3
Revised GUI Sample using AJAX with Color .....	6
Technology Alternatives .....	8





## Preface

This document is the first version for this type of GUI Design. Additional refinements can easily be made to make the design more useful and more efficient as well as a more professional look and feel.

Making this GUI Design live in a real production environment would require a bit of JavaScript with some HTML/DHTML/CSS. Making this GUI Design live with a sufficient level of robustness would require the use of an existing product known as [ezAJAX™](#).

## Problem Statement

Efficient and Useful GUI Designs are those that allow end-users to quickly deal with large complex datasets using the fewest interactions whether those interactions are keystrokes or mouse-strokes.

Consider the problem of presenting a large dataset of an end-user; 20,000 records that have to fit into a listbox or a drop-down combo box. What is the best method for doing this ?

The typical approach most people might immediately think of might be the following:



This is nothing more than a simple selection list coded as HTML. The idea here would be to allow the server to populate this list which may contain 20,000 items or more from which the user would be asked to choose one item. This would truly be a daunting task for any user.

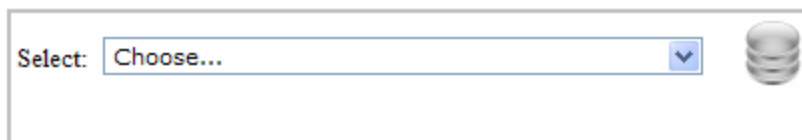
Now what happens if we use AJAX to help reduce the load of having to download 20,000 items that would have to populate the selection list.

## Why AJAX ?

AJAX makes it possible to download and populate smaller chunks of data or content as opposed to having to populate an entire GUI page at a time from the server using more traditional techniques such as PHP or the like.

## Revised GUI Sample using AJAX

Now that we are using AJAX we can revise our GUI Design to look like this:

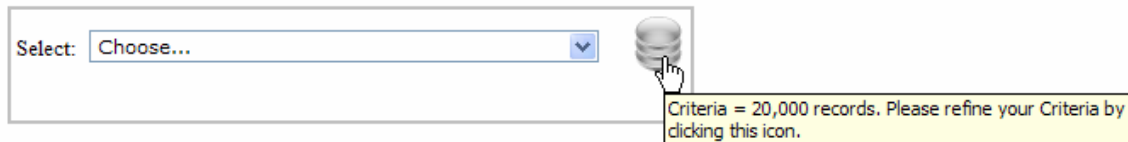


Notice the icon that now appears, it looks like a stack of discs.



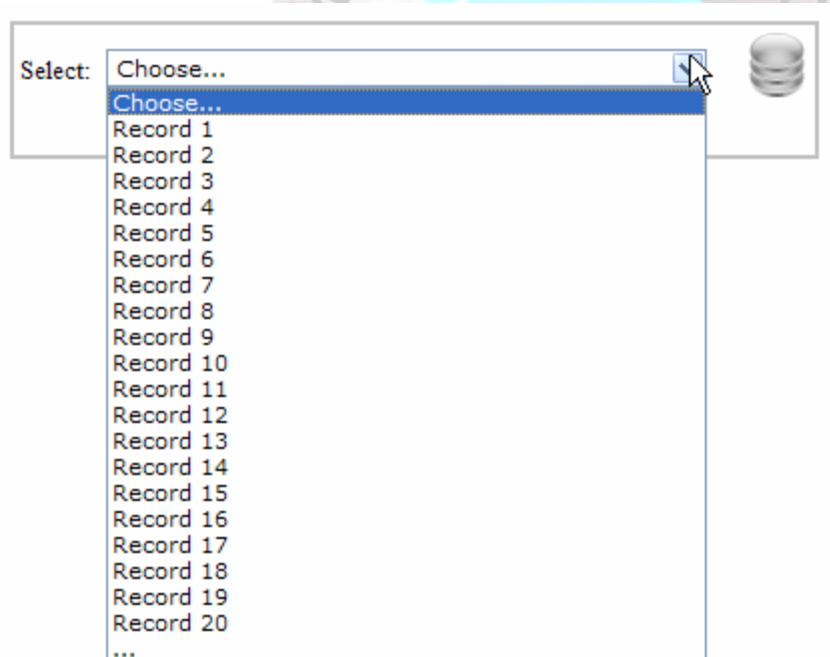


When the end-user hovers the mouse over the stack of discs icon the help text appears briefly.



Criteria = 20,000 records. Please refine your criteria by clicking this icon.

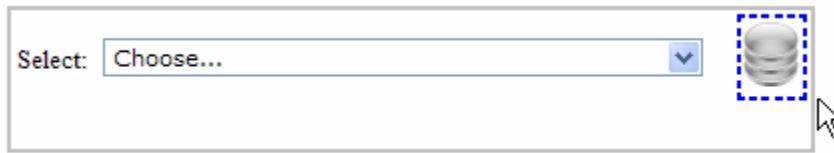
This is a sample message that could be modified to be more meaningful to a particular community of users however the stack of discs icon does convey the amount of data that would be returned unless the Criteria is modified. Notice at this time only the first 20 records have been returned to the client to serve as a sample of the large number of records the end-user would be expected to choose from.



As you may imagine, based on the visual cues you already know about, there are too many records to download; the end-user will notice the “...” that appears at the end of the displayable sample of record that indicates there are more records in the database than can be shown.

So far the current design is fine but now let's add some color to help communicate what is happening to the end-user; colors communicate faster than textual cues.

## Revised GUI Sample using AJAX with Color



As you can see we have added one simple GUI element to the evolving design; the blue dashed border around the stack of discs icon. The blue dashed border would bring the user's focus to the stack of discs icon.

We are using color as a way to convey meaning rather than showing a textual message.

Now we want to draw the user's attention to the specific area of focus, in case there is more than one area of focus on the web page.

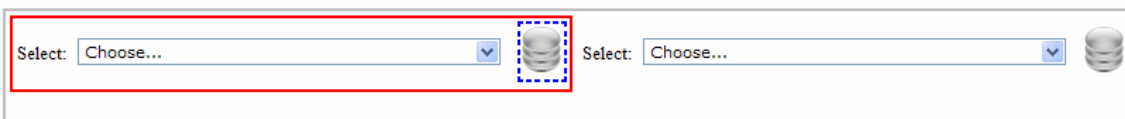
Consider this sample web page:



Can you tell which area of this GUI should have your focus ?

Yes, you guessed it; the area with the **blue** border.

Now what happens if there is an error while the user is interacting with this GUI design ?

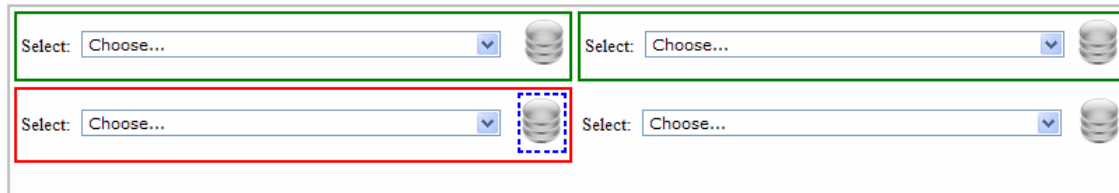


Can you tell where the error may have happened ?

Yes, you got it right; the area that is bordered by the **red** color.



Now what happens if we want to convey the fact that there are no errors in a particular area of the GUI?



Can you tell which areas of this GUI sample that have no errors versus the one that does?

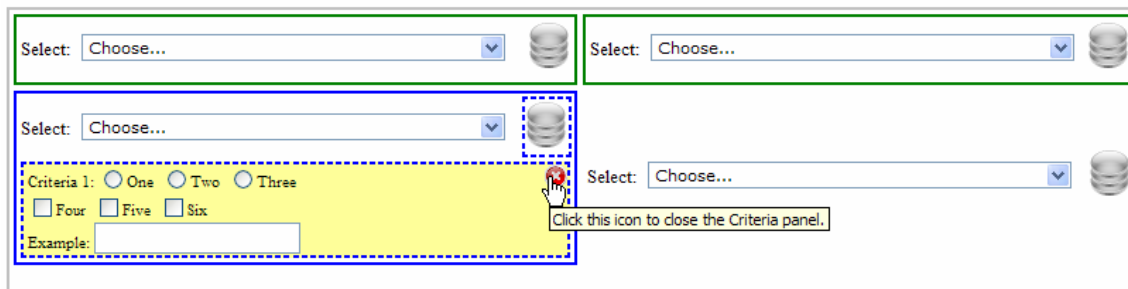
Yes, you guessed it correctly; the areas bordered by **green** means no errors and the area bordered by **red** means there is an error.

By now you should have the feeling that **Red** and **Blue** will never appear on the same GUI at the same time; **red** is the opposite of **blue**.

The dashed **blue** border only appears in the area of focus and only when there is a **red** or **blue** solid border area an area.

We will discuss later on as to how we focus the user on each area and how we coordinate all this color and icon design information.

Now let's show how we would allow the end-user to choose some Criteria to help narrow the search.



Now we can see how the Criteria Panel might appear within the area of focus.

Notice the various Criteria; this panel would be modified to fit the specific scenario for each actual area of focus. AJAX would be used to convey the various choices to the server as the user interacts with them. The stack of discs icon would be updated dynamically as the number of records is narrowed or expanded according to the actual criteria as the criteria is being selected by the user.

Radio buttons indicates one choice of many; checkboxes indicates many choices of many and the entry field indicates a Query By Example that allows for freeform text entry that may contain wildcards or some other expression format such as but not limited to Regular Expressions however it should be noted your typical end-user might not even know how to construct a Regular Expression.

Only one Criteria Panel would be displayed below each area of focus at a time.



Now let's consider how the evolving GUI might appear when the Criteria narrow the number of records to a smaller number than the original 20,000 records.

The screenshot shows a web interface with four search criteria areas, each with a 'Select: Choose...' dropdown and a disc icon. The third area is highlighted with a yellow background and a dashed blue border. A tooltip points to the disc icon in the third area, stating: 'Criteria = 2,000 records. Please refine your Criteria by clicking this icon.' The third area contains radio buttons for 'One', 'Two', and 'Three' (with 'One' selected), and checkboxes for 'Four', 'Five', and 'Six' (with 'Five' and 'Six' checked). An 'Example:' text box is also present.

Notice the various stack of discs icons have been changed to reflect the fact that the end-user has adjusted the criteria for 3 of the 4 areas of focus and that the end-user has just chosen some Criteria for area #3.

This GUI design conveys a lot of information through the use of colors and icons. Refinements can be made to better convey the actual status of the database in terms of the specific choices being made by the end-user in real-time through the use of AJAX.

## Technology Alternatives

AJAX can be used in the form of JavaScript and HTML/DHTML however Flash would allow for a more dynamic display through the use of animation in various areas of the display. Obviously if Flash is being used then the form of AJAX is also known as Flash Remoting or interactions with a Flash Media Server.