

## FORM LAYOUT STYLE GUIDE

Developer note: the conventions shown below are taken from the provisions and assumptions solutions, however, they hold true for all screens. When exceptions to these guidelines are encountered, please notify the IA to create an addendum to these guidelines that cover the UI exception.

### OVERVIEW

The application will consist of many screens where the main activity will be the entry and viewing of many data fields. These fields should be displayed and treated in a consistent manner. The goal of this specification is to set the standards to how a data entry screen is constructed, laid out and common fields and actions used on every screen that exhibits the same functional goals.

### SCREEN DIMENSIONS AND LAYOUT OF LABELS AND INPUTS

The entire screen is specified at 1024 with 2/3 or the screen devoted to the action area, this area is determined to be roughly 800 pixels. The left hand side is given roughly 10-20 pixels of padding to remove it visually from the left hand menu. The remaining area is divided into 4 equal quadrants of 180 pixels each. Between each column is a gutter of 10px. The goal of these standards is to ensure that label-value pairs are read as being the most prominent element in a screen layout.

The left hand most column will contain labels. Labels are 8pt Lucida Sans Unicode right aligned in a container of 180px. Labels are centered text baseline with the input area text. All text in the system is aligned to the text baseline rather than bottom or top of the container.

The next column contains input controls which will be 180px justified. In some exceptional cases it may be necessary to include 2 inputs and extra labels in one column. It is best to work around these sorts of inputs or make them conform to the standard where possible.

When an input has focus, it will have the windows color of 'info'.

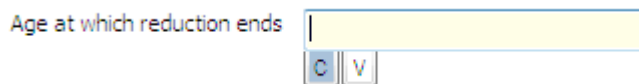


Figure 1 - Input label value pair with focus color showing alignment and width of each element

### HELP AND NOTES

The final 180 pixels column of the screen are reserved for help, notes and utilities. The specification remarks about the need for the user to enter notes on each screen, also, to see help topics about specific areas within the application. It is preferable to do these actions inline, so the final 180 pixels of the right hand side of the screen is reserved for inline help/notes/forum. In the illustration below, the paradigm of MS Office help is used, but this is a placeholder for the desired functionality. The current functionality will be to use tooltips for help when the users mouseover specific labels or headers where help is available. Notes can be both a way to remember ideas when returning to a page, or to facilitate workflow by sharing or even emailing notes. This textbox currently will be

implemented through a popup window, but will later be incorporated inline to both enter and display notes about a page. Notes may also serve as a feedback mechanism to the developers or administrators to note behavior or flow issues within a page. If used in this manner, notes will be generalized about a page, but not filtered per user. There will be a specific use case on help and notes with more details.

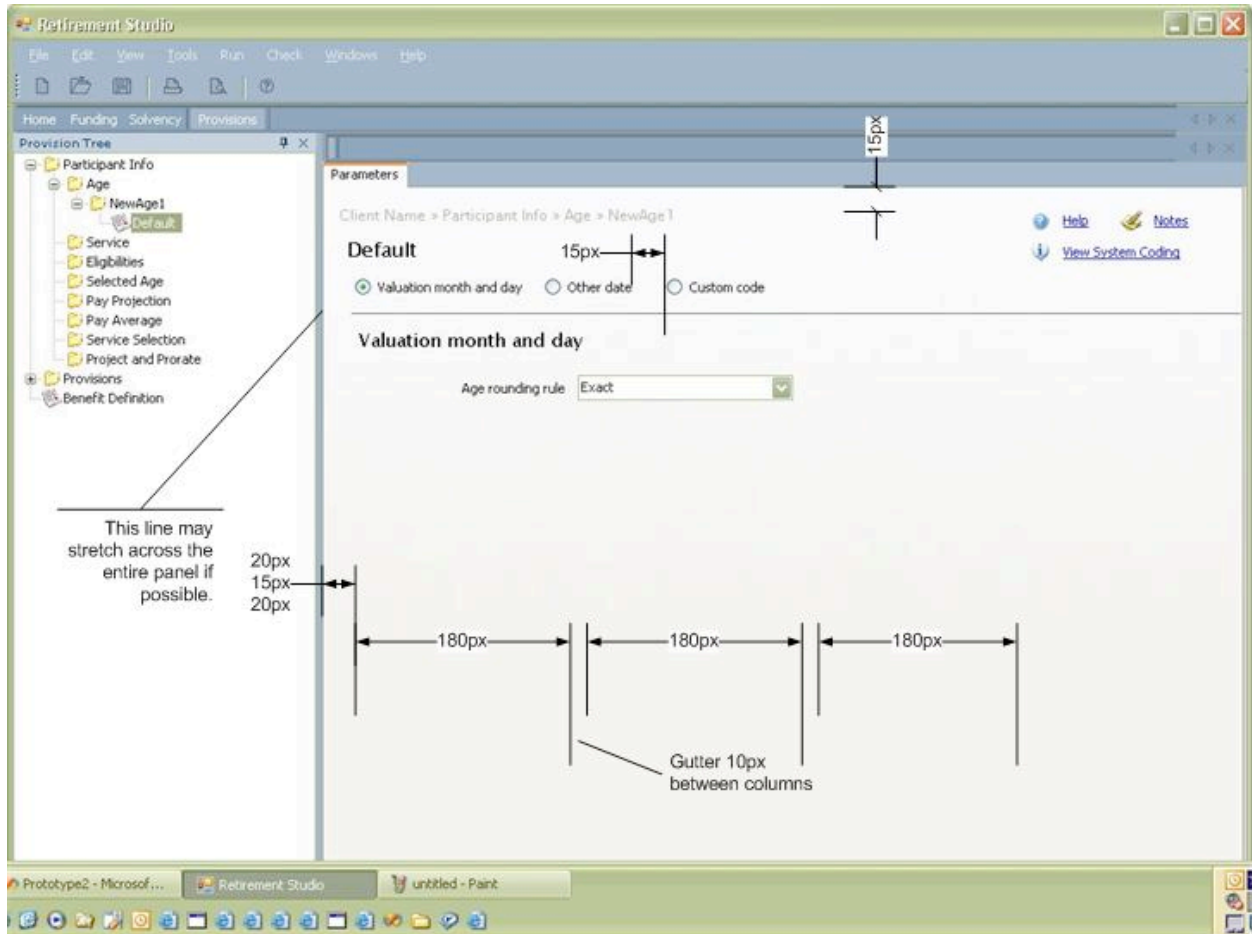


Figure 2 - revision two of layout from prototype

The 'utilities' shown on the top left corner of the page consist of links to help, notes, and coding in parameter screens. Where not required, only the help and notes are universal. The far right hand column underneath these utilities are reserved for help text and note taking. This functionality is forthcoming and will be a style entry when complete. These links open popup windows.

## HEADERS , TABS AND BREADCRUMBS

Parameter pages have an area for tabs (under review) , not changing from existing implementation of tab area at present.

Plan name » Node type » DAAMP node name » Definition name

## NewMortality Decrement1

Figure 3 – breadcrumb specification

The first line which is indented 10 px from left and top contain the breadcrumb trail beginning with the client name, the node name, and the area in focus. This text is Lucida Sans Unicode 10pt 'grey text' in windows. It is not clickable and has an overflow property of 'ellipsis' which will truncate if longer than the page.

The intended information shown is: 'plan name' – 'node type' – 'DAAMP node name' – 'Definition name'

The main header of the page is 12pt Lucida Sans Unicode. The main header shows the name the user has given the particular parameter set.

Secondary headers are 10pt Lucida Sans Unicode. Main headers will be used to represent the options chosen by the user. The heading and the option text should be identical.

The header is separated from the body of the form by a line with 7px padding on top and bottom.

### OPTION GROUPS

The header may also contains option groups, if there are options for the page, they will be arrayed (alignment and label sizing in option groups is not necessary) in a line. If exclusive options are needed, they will be represented by radio buttons. If non-exclusive options are available they are represented by checkboxes. Most 'option' pages require a selection to proceed, so the canvas area will contain a reminder to select an option. Most pages will also contain the ability to bypass the input boxes and go directly to system coding to enter a formula.

Client Name » Funding » Decrements » NewDecrements1

[Help](#) [Notes](#)

#### Default

[View System Coding](#)

Use legislated tables    Same structure for all periods    Pre decrement-pre/post commencement    Pre/post commencement  
 Gender based    Disabled vs healthy    Participant vs spouse

Figure 4 - example of header with option group

Options will 'trigger' display of various inputs below a line, which marks the transition between the header and canvas. This may be reviewed and restyled, but the line is sufficient for now.

Radio button options are used to indicate a terminal decision, the user can only choose one of these particular options to display choices, however, in some pages, there are 'ala-carte' options. They can be combined with most but not all radio choices. They are represented on two lines and should not be co-mingled if possible. See 'variations' for examples of how to handle complex option/'ala-carte' groups.

### VARIATIONS

---

## COLUMN INPUT USE

The second column for inputs will be utilized when a comparison or division of one input is presented to the user. In this case there will be one label, and a pair of inputs that are identical. In these situations the UI will also present column headers which are 9pt MS Sans Serif.

The image shows a UI element with a checked checkbox labeled "Use gender based". Below it, there are two columns. The left column is labeled "Male" and the right column is labeled "Female". Under the "Male" label, there is a text input field with a "Constants" label to its left and two small buttons labeled "C" and "V" below the input. The "Female" column has an identical structure with a text input field, a "Constants" label, and "C" and "V" buttons.

Figure 5 - 2 column input example with trigger

---

## INLINE DISABLING OPTIONS

It may be necessary to trigger enabling or disabling of controls within the parameter action panel, when this is required, the default behavior will be to 'grey-out' or disable controls rather than showing and hiding controls.

This becomes more difficult when labeling is necessary as in figure 5, in this case, triggering gender both adds a row and creates a column header. In this case showing and hiding the controls is the preferable method, although revisions for consistency sake may be required.

---

## INPUT MASKS

To aid in user input, error correction and validation, many fields will include an input mask,

---

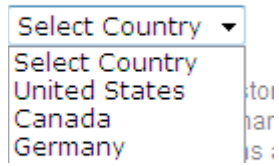
### NUMERIC – PERCENTAGE

---

### NUMERIC – INTEGER

---

### TEXT - DROPDOWN



When the application calls for a finite list (up to 50 elements) the system may present these finite options within a dropdown list. This list is traditionally sorted alphabetically, however in some cases there is a practice of using option grouping headings to demonstrate different categories of options, as well as the practice of putting the most commonly used options in the beginning of the list, separating the rest of the list by a line and proceeding alphabetically.

