sinimuna

Sinimuna UX Design

WMS Style Guide for
DataScan Technologies

August, 2006

UI Styles & Standards Guide

Version 2.0

Last Update: August 22, 2006

UI Styles & Standards Guide

Created by Lisa Hansen

Maintained by Lisa Hansen, Nick Zedlar and Maggie Rivera

Table of Contents

Introduction

Function

This standards document is designed to support and drive user interface (UI) design and front-end development tasks. In the absence of further clarification from a live person authorized to address UI issues, this document should be considered the final reference authority on these matters.

Authority

This means that the guidelines herein are not merely recommendations; they are standards. Follow these standards in all cases, without deviation, embellishment or selective application. Only rarely – under specific circumstances and only with the approval of UI design, business analysts and/or product owners – are exceptions allowed. This authority derives directly from management.

Rationale

Standards serve a vital purpose in maintaining consistency. Since a learning curve is associated with all software, consistency conditions the user to expect certain predictable patterns in both presentation and functionality. Similar instructions, for example, worded in similar ways, appear in the same font face, color and size in similar screen locales under similar circumstances.

Furthermore, with consistency, the company's image and brand are perceived as a coherent whole. This in turn conveys stability and assures the client that their trust in the company's products and services is not misplaced. It fosters loyalty and cultivates an emotional bond with the brand.

Evolution

This is an evolving document. The mutually symbiotic relationship between this Style Guide and its associated products mirrors that which occurs between form (art) and function (application). Both are engines of change, recursively refining, improving and maturing.

The prime objective in this endeavor is, of course, to produce the best quality software on the market. To that end, change is a vital, and by definition necessary, component of betterment. The constantly fluid nature of software development necessitates that these guidelines variably act and react to new conditions.

Therefore, the reader is admonished to consult this document on a regular basis to remain up to date on revisions, additions and clarifications. Everyone affected by changes in this document must remain current on all standards.

Browser and Keyboard

Browser Requirements

By design, Falcon applications launch in their own browser pop-up windows. In order to operate in a known environment with prerequisite capabilities, they assume control of numerous native browser behavioral and functionality features. For instance, prior to full application start, a log-in message instructs the user to disable pop-up blockers or use **CTRL+Click** at launch.

Falcon applications are designed to function with the Microsoft Internet Explorer 6.0 (MSIE) browser only. Cross-browser compatibility is not presently in scope. In most cases, functionality is ensured with the default browser settings plus the following:

ENABLED: JavaScript, cookies, proxy server and 128-bit encryption

DISABLED: pop-up windows and third-party plug-ins, extensions and add-ons

Toolbars and Address Bar

Launch coding eliminates the address bar and extraneous toolbars.

Full Screen Window

Employ when possible a maximized or “full screen” window to ensure that the user benefits from the largest viewing area.

Scrollbars

Scrollbars, both horizontal and vertical, must remain enabled to allow movement within the window. This is particularly relevant to tables with numerous rows and/or columns that do not fit within the confines of the application window.

Status Bar

Via interaction with the server, the status bar communicates the application's real-time processing state to the user. If intentionally disabled, the application requires some alternate progress indicator.

Right Mouse Button

Along with restricting toolbars and the address bar, disabling right mouse clicks prevents users from navigating via the browser's "Back" and "Refresh" in the right-click menu.

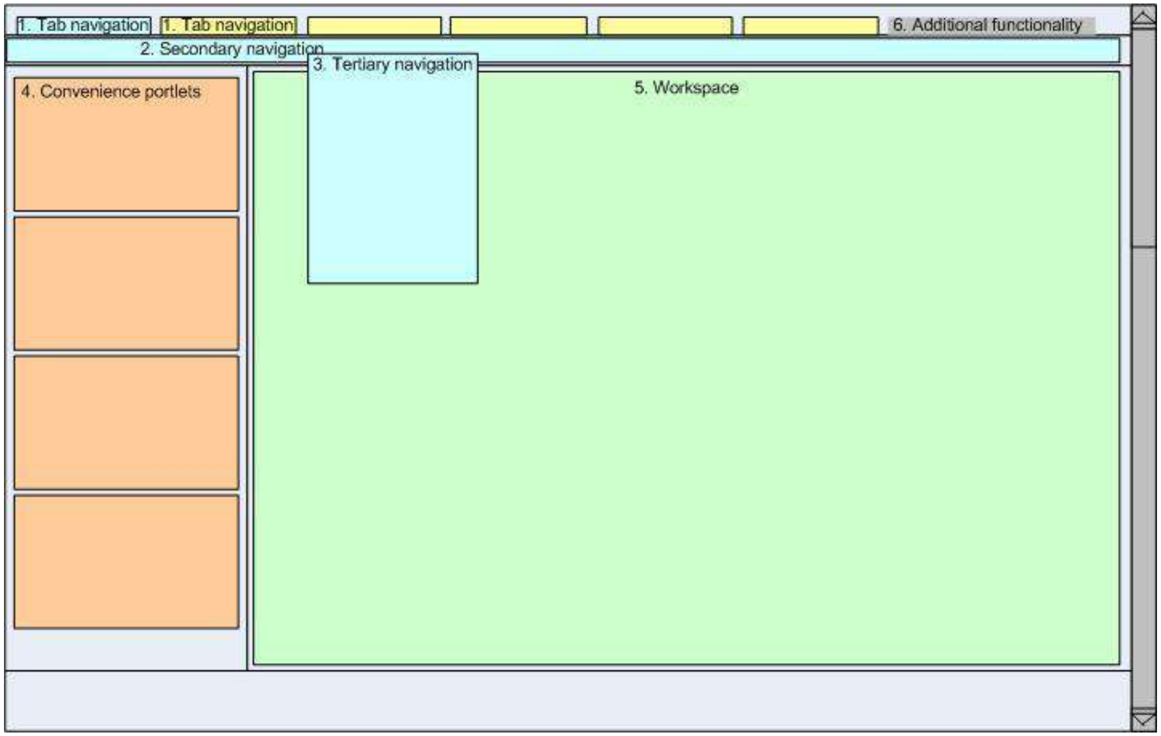
Hotkeys/Keyboard Shortcuts

The current WMS system supports full keyboard-driven navigation. Inherent limitations of Web applications, such as poor support for keyboard-driven navigation schemes, necessitates selective implementation of this functionality.

Data entry-intensive interfaces (such as those used for transactions) receive their required keyboard navigation support at the task (portlet) level. Keyboard-only interactive tasks have keyboard equivalents for all task-level elements. Observe careful attention to tab order.

Certain frequently accessed items, such as the query builder, specific reports and review screens, receive keyboard shortcuts. To support this functionality, a convenience portlets provide item short lists.

Portal User Interface



Navigation

- ***Primary (Tabbed)***

Tabs are stateful. They must be carefully structured to allow for best use of stateful tabs for multitasking (for instance, have both a payments and a loans tab to allow both tasks to be active at once). Or, allow secondary navigation to be stateful. Moving to another tab triggers a Save prompt if user has entered unsaved data into a form.

• *Secondary*

May need to be stateful to allow for best multitasking (may not be able to have enough tabs to support this need). These are not tasks, and should not be clickable. Instead, they trigger the tertiary navigation shown in a rollover. Onmouseover shows tertiary navigation list.

• *Tertiary*

Rollover navigation: Each item represents a task (or possibly a set of tasks) to be launched. The item may also be a menu of related items that should be done in listed order (i.e., the entire setup for GL can be a single menu item that, when launched, displays a menu of all included tasks.

Moving to another task triggers a Save prompt if user has entered unsaved data into a form. A task selected under the same tab replaces any active task on the screen.

Portal Function

• *Convenience Portlets*

Provide various portlets that are dynamic (recent tasks, frequent tasks) or user-managed (favorite tasks, top 10 reports). Need a variety of these; may depend on the tab.

Would like to have a portlet that tracks the open task across tabs. Task name should be clickable to allow quick access to that particular screen. Similar in nature to breadcrumbs, but it crosses tabs and consolidates info into a single space.

Example: Main>None; System Administration>Add Employee; Markets>None; Transactions>Pay Loans; etc.

Each convenience portlet should have show/hide that reduces the portlet to show only the header element. Also, these portlets could have ‘X’ close function, hiding the portlet completely until next log-in.

To do: Determine how convenience portlets will be created. Are they based on user behavior collected by the portal (for instance, to dynamically build a list of recent tasks) or are they user managed, which requires Edit capability to build a favorites list.

- ***Workspace***

Where selected tasks are displayed.

Each portlet has a "default" task. In some portlets, the "default" task contains minimal content, perhaps only a line of text that says 'Select a menu item.' In other portlets, such as Reviews/Reports, the "default" task is a menu of tasks available in that portlet.

Need to be able to string tasks together (i.e. all the tasks for GL setup, or all the tasks for Customer setup) in a wizard format, in order to reduce incomplete setup work. This is outside Portal functionality; will have to be tackled at the task/UC level.

Will contain a single active portlet. Users were informed that if we provide sets of portlets, need to know in advance what those portlets are.

'Closing' a portlet returns the screen to the 'Tab Main' state.

Must be able to maximize/minimize primary workspace.

Moving to another tab triggers a Save prompt.

A second portlet below the Workspace may be used to contain a Reviews/Reports portlet. Not determined – whether this Reviews/Reports portlet will launch in a new window. Using this portlet should not require the user to save work in the Workspace portlet.

Investigate the Following…

Need to determine whether multiple tasks can be strung together in a wizard approach (may not be in Phase 1).

Need final decision on how Reviews/Reports portlet will work – new window/same window.

Launching and closing tasks

Each portlet has a "default" task. In some portlets, the "default" task contains minimal content, perhaps only a line of text that says 'Select a menu item.' In other portlets, such as Reviews/Reports, the "default" task is a menu of tasks available in that portlet.

When a task is launched from the menu or a convenience portlet, it is launched into the portlet defined to display it.

If a menu-selected task (not the default task) is already active in the same portlet, the newly selected task replaces the existing task. If the existing task has unsaved data, the user is prompted for unsaved changes.

When a task is launched from another task (parent-child related tasks), it overlays the task from which it is launched, in the same portlet frame.

Closing a parent task returns that portlet back to the "default" task.

Closing a child task returns the user to the parent task (concept- "backing out" of nested tasks). This is true regardless of how many nested tasks there are – clicking close returns the user "back" one task. The "Submit" button always keeps the view within that portlet. Only "Close" will take the user out of the task and back one task.

- ***Additional functionality***

Print – prints whole screen (replaces print command in browser buttons)

Print function on individual portlet frames. Want ability to print only the contents of an individual portlet frame.

Text size up/down – changes font size (replaces font change command in browser buttons)

Help – triggers global help function

Log out – log out of system

Investigate the Following…

Need to review what additional functionality we want included here.

Need to determine how global help is triggered – new window? Same window?

Need determination on ability to print individual portlet contents.

Whole browser window scrolls, and tables within portlets scroll, but portlets do not scroll.

Footer contains copyright and terms information (standard footer info).

General Formatting

Cascading Style Sheets (CSS)

A cascading style sheet (CSS) is a list of customizable styles a browser will use to present Web page content. These typically define relative and/or absolute attributes from the document level down to the individual element, regarding among other things font face, size, decoration, indentation, wrapping and color; button, field, table and table data cell background colors, images, padding, spacing, vertical and horizontal alignment, border thickness, color, dimensions and type; and a host of other presentation layer controls. All Falcon pages reference external CSS styles. Ones embedded in the <HEAD> or occuring inline in individual HTML documents are strictly forbidden.

Deprecated HTML

Avoid outdated legacy, or deprecated, HTML tags and attributes at all costs! Use CSS without deviation, so that changes apply universally. The advantages are greatly simplified style changes and consistency across the application. Examples of deprecated code are as follows:

- <TABLE BORDER=”1” CELLPADDING=”4” CELLSPACING=”2” BGCOLOR=”#FFFFFF”>
- <FONT FACE=”ARIAL” COLOR=”#OOOOOO” SIZE=”12”>
- <B>, <I>, <EM>, <CENTER>

No absolute font sizes should be used. Use a percentage value when determining font sizes, in order to maintain relativity of size and to promote accessibility.

Tables

Nested tables are inevitable, but use the fewest possible to attain the desired look. Do not appy absolute widths or heights to tables or their components. Percentages may be used, if needed, but must be defined in the CSS.

In tables containing more than one column, a header should be provided that indicates the contents of the table. No header is needed for tables containing only a single column.

Tables used for selecting data for population into the primary form.

These tables typically appear in grey to the right of (occasionally below) the primary form.

In order to select a line item to populate data into the form, the first column in the table contains data associated to a hyperlink that, when clicked, populates that data into specified elements of the primary form.

Only the first column contains links, unless otherwise specified.

Empty Tables. Certain tables may contain no data. Below describes how to display empty tables.

In tables generated by a search, the table is not shown until a search is conducted. If a search contains 0 results, the table should not be displayed. Instead, use an error message to indicate "No results found."

In tables not generated by a search, the table is always shown. If the table contains no data, an indicator stating "No results found" should be provided. This indicator is either shown in a blank row in the table, or as an error message immediately above the table.

- ***Alignment***

Right align all numbers (except dates) in tables

Center align dates in tables

Left align text in tables

- ***Sorting***

Tables are always sortable unless otherwise specified by the use case or functional case documentation.

Tables are sortable by clicking the column header; clicking the column header a second time reverse-sorts.

Tables are default sorted by the first column in ascending order, unless otherwise specified. The Business Analyst may always choose to specify something other than the default sort.

Clicking a different column to sort triggers an ascending order sort. Click the same column a second time to descend-sort.

Sorting is case-insensitive. For instance, a=A.

Lists can be sorted in ascending and descending order. Ascending sorts have an upward pointing arrow. Descending sorts have a downward pointing arrow.

Null or blank values are treated as the first items in an array sorted in ascending order. See sort order for column values, below.

	Numbers	Alphanumeric	Dates
Ascending	low to high (null, 1,2,3,4,5)	low to high (null, a,b,c,d,e)	earlier to later (null, Jan1, Feb1, Mar1)
Descending	high to low (5,4,3,2,1, null)	high to low (e,d,c,b,a, null)	later to earlier (Mar1, Feb1, Jan1, null)

Copy/Text

• *Capitalization*

Capitalize All of the Key Words (title style)

Page headers

All task titles

Form field labels

Tabs

Example: Manage Loans

Capitalize only the first word and proper nouns (sentence style)

Secondary menu items

Buttons

Screen instructions

Error messages

Any longer text elements

Example: Apply for a loan

- ***Spelling***

Standard U.S. English spelling is observed. More detail will be added here, as time allows.

- ***Hyphenation***

Commonly hyphenated words include: e-mail, mock-up, pre-computed, non-serialized. If in doubt, hyphenate.

Pagination

Unless otherwise specified in the UI and wireframe documents, no pagination will be used for lists.

Only lists that are long and content-rich should include pagination (such as in Comments, Queue items, audit histories, and review screens).

Paginate in sets of 25 (i.e., 25 results per ‘page’)

The current page number should not be clickable, and should be bold-face.

Pagination should not appear on lists that are only one page long (less than 25 items).

'First', 'previous', 'next', and 'last' elements can remain active regardless which page the user is viewing.

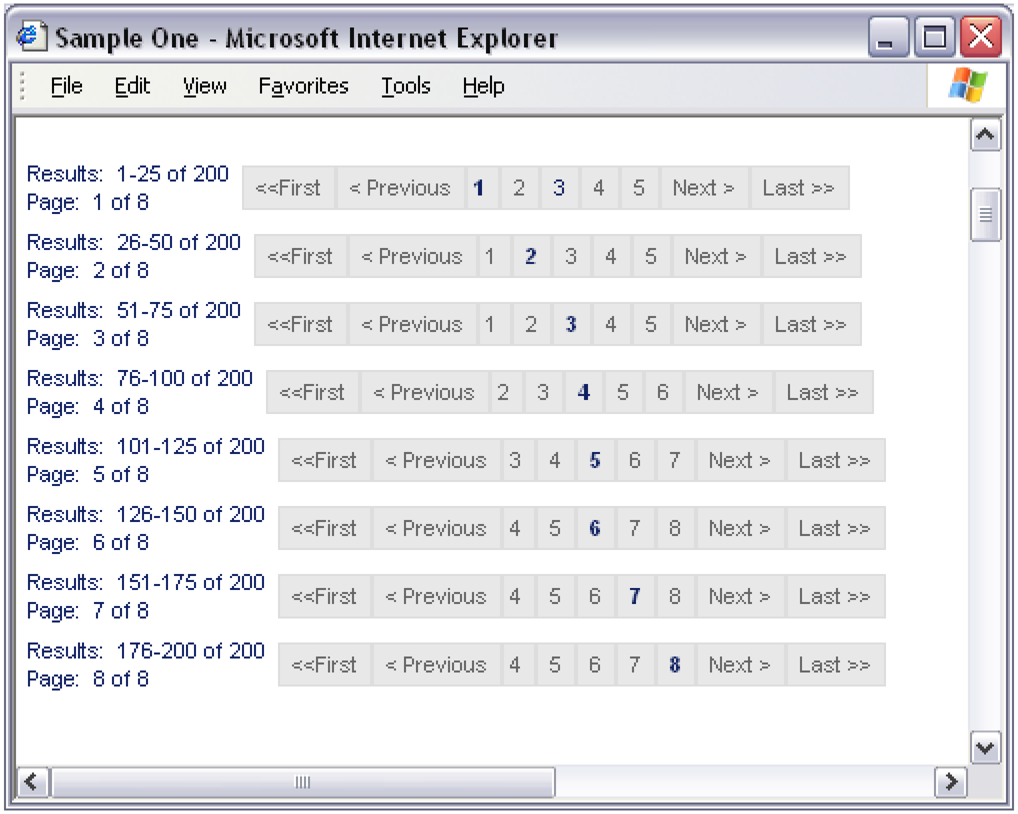
Summary of links:

Link	Description
<<First	User moves to the first page in the list.
<Previous	User moves one page back from the currently viewed page
Page numbers (1, 2, 3, etc)	Move user to that page. As user moves through the pages, the active page number is not clickable and is in the center of the number array. The only exceptions are the first two and last two numbers in the array, which sit in first, second, last-1, last positions respectively. If the number of pages is less than five, show only those page numbers.
Next>	User moves one page forward from the currently viewed page
Last>>	User moves to the last page in the list.

Summary of information:

Link	Description
Results #-# of total#	indicates which section of 25 items are being viewed, and the total number of items
Page: # of #	indicates which page is being viewed, and the total number of pages

Example of how pagination appears in an 8-page, 200-item list:



Results: 1-25 of 200
Page: 1 of 8
<<First < Previous **1** 2 3 4 5 Next > Last >>

Results: 26-50 of 200
Page: 2 of 8
<<First < Previous 1 **2** 3 4 5 Next > Last >>

Results: 51-75 of 200
Page: 3 of 8
<<First < Previous 1 2 **3** 4 5 Next > Last >>

Results: 76-100 of 200
Page: 4 of 8
<<First < Previous 2 3 **4** 5 6 Next > Last >>

Results: 101-125 of 200
Page: 5 of 8
<<First < Previous 3 4 **5** 6 7 Next > Last >>

Results: 126-150 of 200
Page: 6 of 8
<<First < Previous 4 5 **6** 7 8 Next > Last >>

Results: 151-175 of 200
Page: 7 of 8
<<First < Previous 4 5 6 **7** 8 Next > Last >>

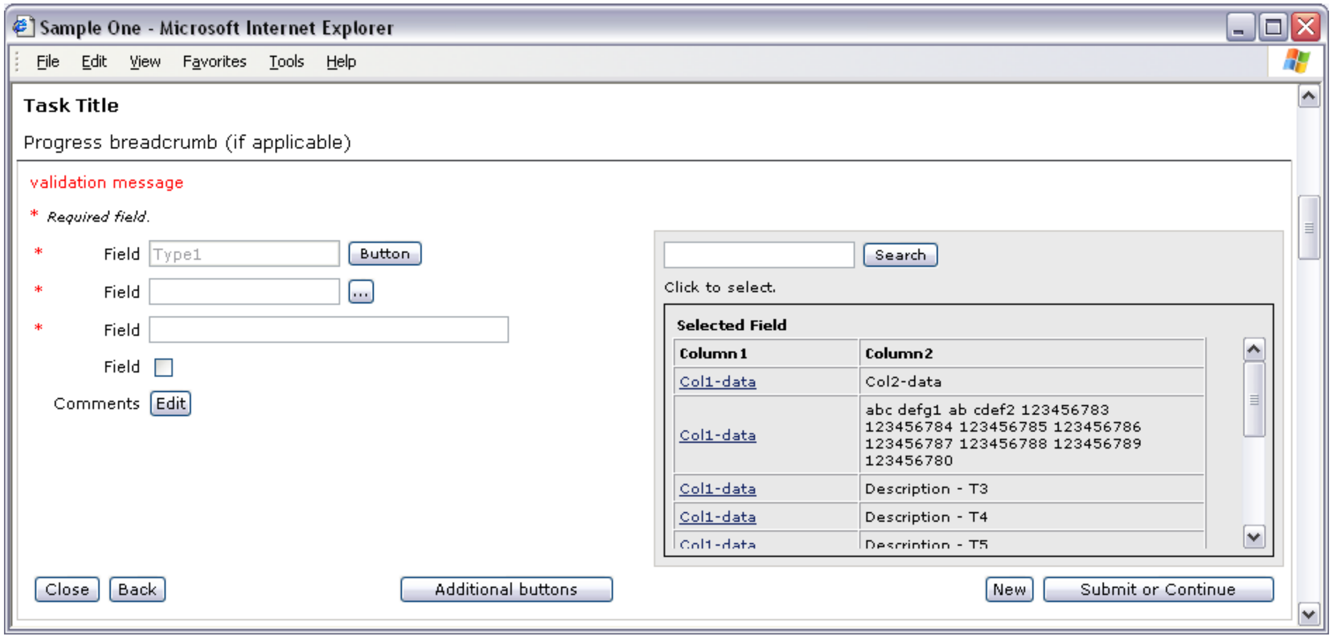
Results: 176-200 of 200
Page: 8 of 8
<<First < Previous 4 5 6 7 **8** Next > Last >>

Forms

Layout

Standard forms include vertically aligned form entry elements on the left. To the right, additional space (shown in gray, below) is used for look-ups and searches pertaining to the form.

Basic structure of a form layout:



Task Title

Progress breadcrumb (if applicable)

validation message

* *Required field.*

* Field Type1

* Field

* Field

Field □

Comments

Click to select.

Selected Field

Column1	Column2
Col1-data	Col2-data
Col1-data	abc defg1 ab cdef2 123456783 123456784 123456785 123456786 123456787 123456788 123456789 123456780
Col1-data	Description - T3
Col1-data	Description - T4
Col1-data	Description - T5

Labels

All field labels sit to the left of the input field.

Field labels are right aligned.

Input fields are left aligned.

All field labels are capitalized (key words): This is a Standard Label

Labels next to clickable interactions (radio buttons, check boxes) must be clickable also. In other words, the user may click the label or the interaction element to trigger the element.

Elements

• *Combination Boxes (Dropdowns)*

Values in combo box menus are to be pulled from the code tables. Values seen in UI design documents (i.e. HTML concepts) should be treated as samples. The codes will be documented in the business analysis documentation and any necessary values entered into the code table for use with the form field in question.

Values should be in the following order, unless otherwise specified in the use case documentation.

Blank (the default value unless otherwise specified)

Specific values that are specified in the use case documentation to be forced to the top, shown in alphabetical order (unless otherwise specified) by language (for instance, “All,” “United States”)

All other values, shown in alphabetical order by language

• *Check Boxes*

Use check boxes to select or unselect a single item (see sample above)

Use check boxes next to each line item in tables to allow multiple items to be selected.

Use check boxes next to a column header in tables to allow all items in a list to be selected.

Check boxes shown in a table header row are used to select or unselect all of the items in the list shown in the table.

If an unchecked header row checkbox is selected, all of the list items' individual checkboxes are then checked, regardless of prior setting.

If a checked header row checkbox is deselected (unchecked), all of the list items' individual checkboxes are then unchecked, regardless of prior setting.

• *Mover Boxes*

When data are moved between mover boxes and are submitted, the mover box contents should refresh to show in alpha-numeric, case-insensitive order. Though used in early development, mover boxes are now considered deprecated.
Text input fields

Alignment

Right align

Currency input fields

Any other numeric input fields

Left align

Alphanumeric fields

Comment fields

Center align

• *Text Fields*

Date fields

Use virtual wrap on longer fields.

Numeric fields

Dates

Users should be able to enter MMDDYYYY or MM/DD/YYYY or MM.DD.YYYY or MM-DD-YYYY in a date field. The system should parse this to a standard date format: MM/DD/YYYY. This will change with internationalization (i.e., a European standard may be DD.MM.YYYY).

Currency

No currency symbol should be displayed.

Display: 999,999,999.99

Other numbers

Rates and percentages: 99999.99999

General numbers: 999,999,999

Stock, invoice, serial numbers (which may be alphanumeric): 999999999

Required fields

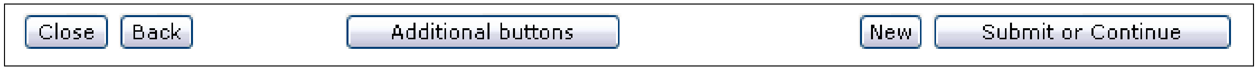
Required fields are designated with red asterisk.

On forms containing required fields, the message * Required Field will be present above the form.

Buttons

Only first word capitalized: This is a standard button

The standard buttons for a portlet task are arrayed across the bottom of the space:



Buttons that are not usable at a certain point in the task (such as Delete in an add task) will be visible but disabled. If, based on a user selection, a button becomes relevant to the task (for instance, a user changes from an add to an edit task), the button will become enabled and active.

Other buttons are interspersed within tasks. See the layout screen example, above, under Layout.

Actions

Close — Closes and cancels the task. In the portal, it returns to a "default" task specifically defined for that portlet. Default tasks can contain minimal content (like an instruction to Select a menu item) or a menu of tasks, such as in Reviews/Reports.

Back — Moves user back one screen (or one state of the screen).

Delete — Deletes the active record shown in the displayed form.

New — Triggers the 'Add' task.

Submit — Submits and saves data to the database, but does not close a task. Form remains populated and the record remains active in the form until the user clicks [**Close**] or [**New**], or selects a different item to edit.

Search — Clicked after a user enters search parameters.

Go — Updates screen/view based on combo box list item selection. This button's enable status is not dependent on whether a selection of a combo box list item has occurred. It is never disabled.

Change — Returns user to screen/view in which the [**Go**] button is displayed

... — Opens or closes the look-up event. If multiple [**...**] buttons exist, clicking a second [**...**] closes the first look-up.

X	Clears the data displayed in a read-only text field that has been populated through a look-up.
Continue	Used for intermediate steps in a multi-screen task.
Edit	Used to trigger a related task element that is not visually represented on the screen.
Calculate	Runs a mathematical routine, usually to populate other related fields.
Log in	Used on the log-in screen only.
<< <	Transfer Left (all items and single item, respectively); used in mover boxes (now deprecated)
> >>	Transfer Right (single item and all items, respectively); used in mover boxes (now deprecated)
Update view	
Refresh	Refreshes the view of the list button is associated to. Used primarily in tasks where a list needs to be refreshed from the database more frequently than upon Submit of adds or changes to the data (see Manage Queue Items).
▦	Launches a date picker pop-up window for the user to choose a day from a calendar application.

Errors and System Messages

Error messages will be provided to developers by product owners, business analysts, and/or documentation specialists.

Error messages appear in context, meaning they are presented within the framework of the task in which the error occurred. Error and system messages appear at the top of the screen or window in which the error occurs, or they appear immediately above the space where the input error occurred.

When an error occurs, the contents of the form fields remain present, both in the error and non-error fields.

Formatting - message

Each set of error messages should begin with the error icon: ❗

Multiple errors should be listed with a <BR> separator.

Formatting – error icon

An icon will be placed next to the field containing the error.

The icon will appear immediately to the right of the field label.

Errors should be as concisely worded as possible.

Messages should specifically describe the nature of the error, to the point that it is clear why an error message appeared. For instance, the statement “The entered value is invalid” does not provide enough detail to prevent additional error. Instead, be specific, using the field label in the error message: “The franistan amount cannot exceed 100” or “The gloxpar percentage must be between 1 and 100.”

Non-error (validation) messages should be handled the same as an error, but without the exclamation point icon.

The class "validation-message" should be used to format error messages. Or, the styles applied in that class should be applied to the appropriate Weblogic class for formatting.

There should be no different background color behind or border around system messages.

All client-side error messages are presented to the user before anything is submitted to the database. These errors should be presented in the order they appear on the form.

Database errors, due to Oracle system constraints, are presented one at a time as they are found. For example, a duplicate primary key violation is found and an error is presented. Then, upon the next Submit attempt, a second duplicate key is found. These errors are presented consecutively and must be resolved one at a time by the user before moving forward to the next server-side validation check.

In order to reduce errors given to the user, fields that require upper-case alpha characters should automatically convert lower-case inputs to upper-case upon submit.

On occasion, validation messages appear as a popup JavaScript message, using the standard OK/Cancel as choices to continue or cancel. This popup message appears as a standard JavaScript message "on top of" a fully rendered display of the Web page from which the user triggered the message event. In other words, the Web page should never be blank; instead, it should always contain the full display the user had just seen and used prior to the popup appearing.

Search

A basic search input field searches from the entire collection of items. Unless otherwise specified, all searches are “contains” searches and search on all fields.

Boolean

- ***OR***

The more information provided, the broader the search will be. This means that the search is an “or” search. For instance, Status=active OR date range=01/02/05-01/05/05 OR loan information=complete.

- ***AND***

The more information provided, the narrower the search will be. This means that the search is an "and" search. For instance, Status=active AND date range=01/02/05-01/05/05 AND loan information=complete.

AND searches are standard. In searches with a single text entry field, multiple criteria should be separated by a comma. If multiple criteria are entered and are not separated by a comma, the system should treat them as a single search string. For instance, if the entered search criteria are: ‘roswell, Toyota,’ the system will search for records that contain both the term ‘roswell’ and the term ‘Toyota’. If the entered search criteria are ‘roswell toyota’ then the system will treat this as a single search string and look for records that contain the text string “roswell Toyota.”

Search interfaces that contain multiple inputs (for the purposes of filtering) are considered AND searches. Most of these searches use multiple input fields; the user may restrict search results by providing more values in the multiple inputs.

In a basic search, each time a search is conducted, the system searches the entire collection of items. Conducting the search a second time repeats the initial search of the collection. In other words, unless otherwise specified, multiple searches with the same interface are not “search within results” type searches.

The requirements for specific fields will vary from search to search. For instance, sometimes a date range will be required, other times it won't be.

Date Ranges

Date ranges may sometimes be required.

Date ranges may sometimes be pre-populated with a commonly-used date range (for instance, today’s date in both the From and To fields).

Sometimes only half the date range is required or needed.

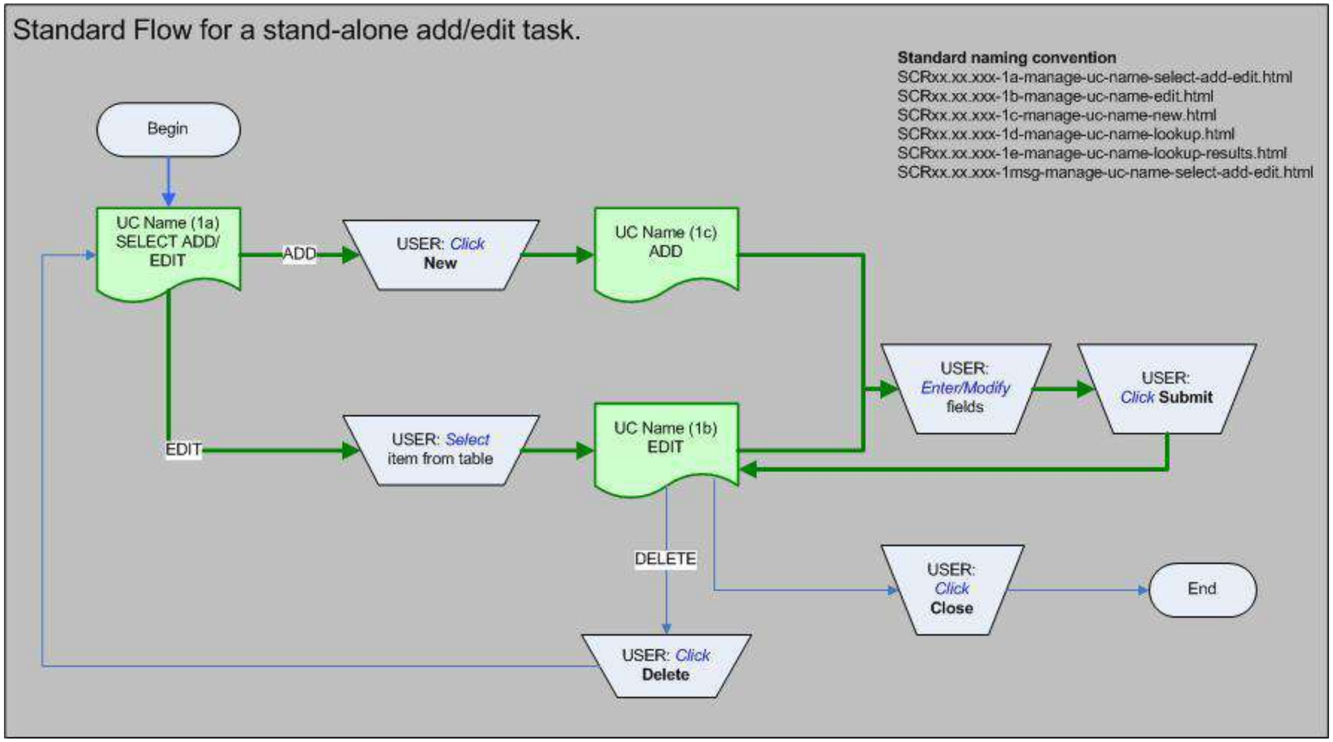
A missing 'From' date and present 'To' date will equal a search for everything up to the 'To' date.

A present 'From' date and missing 'To' date will mean search for everything from the 'From' date forward.

A date range works as expected: 'From' date must be earlier than 'To' date.

Flows

Add/Edit (Maintain)



Standard Flow for a stand-alone add/edit task.

Standard naming convention
SCRxx.xx.xxx-1a-manage-uc-name-select-add-edit.html
SCRxx.xx.xxx-1b-manage-uc-name-edit.html
SCRxx.xx.xxx-1c-manage-uc-name-new.html
SCRxx.xx.xxx-1d-manage-uc-name-lookup.html
SCRxx.xx.xxx-1e-manage-uc-name-lookup-results.html
SCRxx.xx.xxx-1msg-manage-uc-name-select-add-edit.html

1a – The user selects to follow an Add or Edit path, by either clicking 'New' (add) or selecting an item from an active list of items (edit).

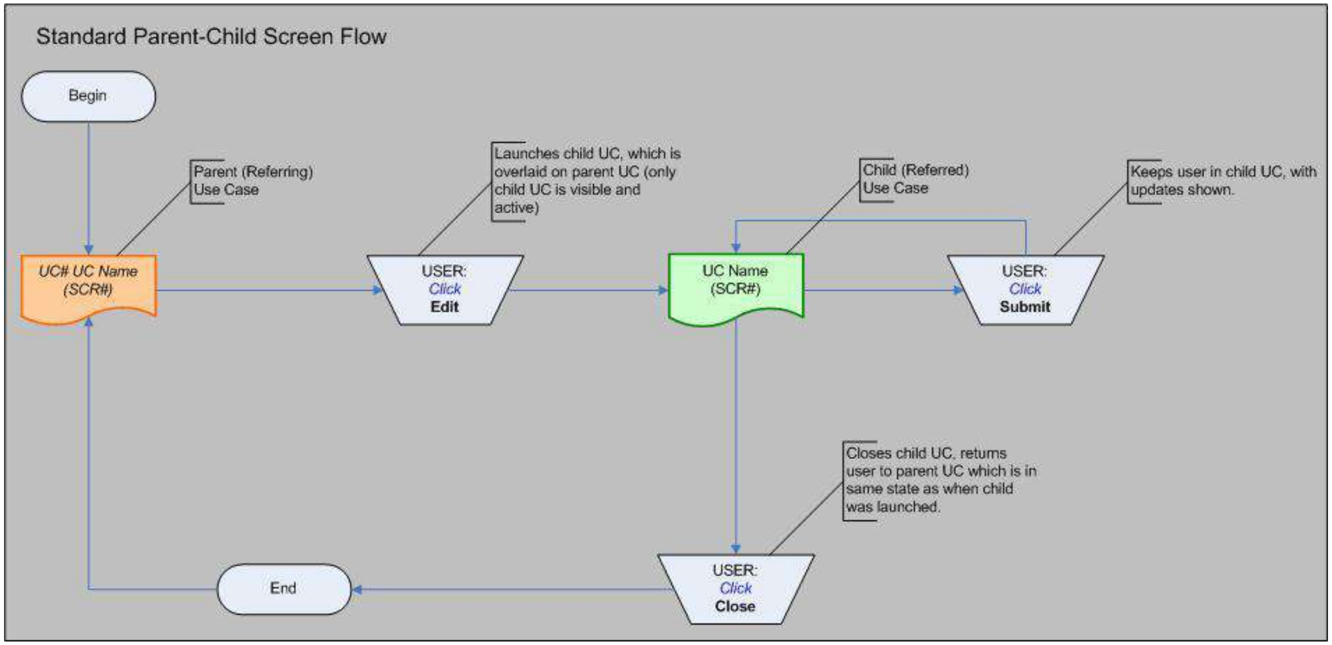
1c – first screen of the Add path. User enters data into the form and clicks Submit, at which point the user is transitioned into an edit mode, with a validation message indicating submission of the record. The record just added remains active in the form, and all fields that are editable in an Edit mode remain editable.

1b – first screen of the Edit path. User modifies data in the form and clicks Submit, at which point the user is shown a validation message, but remains on the edit screen.

From the Edit screen, the user can click "Delete" (if available), at which time they are returned to an earlier state, given that they have just deleted the active record. The record is no longer shown in the form or in the list of editable items.

From any screen, the user can click "Close." If the user has entered or modified any data prior to clicking Close, they are prompted that unsaved changes will be lost if they continue. Only the Close button will take a user out of a task.

Parent-Child Flows



Parent (Referring) Use Case

Launches child UC, which is overlaid on parent UC (only child UC is visible and active)

Child (Referred) Use Case

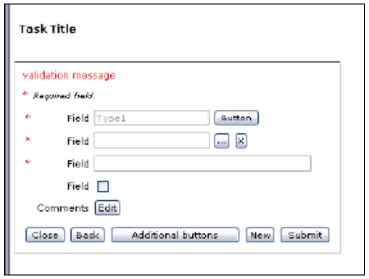
Keeps user in child UC, with updates shown.

Closes child UC, returns user to parent UC which is in same state as when child was launched.

Show and Hide Look-up/Selectable Table

Primary Flow

This is a standard starting point for many use cases which contain a look-up interface for populating certain form data. In this case, the second field on the screen uses a look-up function (triggered by the "…" button) to populate the form field. The text field is read-only; the user must select what data to display there. In addition, to clear that form field, the user must use the "X" button.



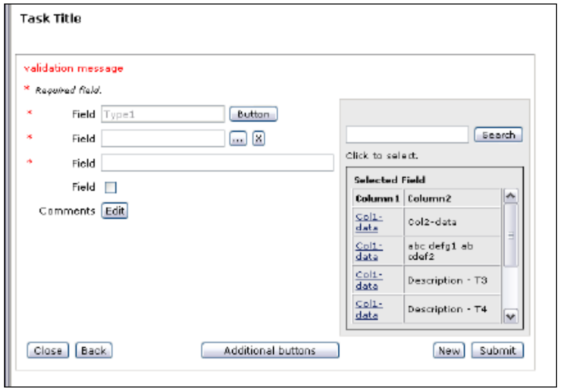
Task Title

validation message

* Required field.

View 1

Clicking the "…" button triggers a screen display change to show a searchable and/or selectable look-up form for the associated field. Note: depending on the use case, the contents of the table may not be present until a search has been made. In other use cases, there may be no search interface, just a table.



Task Title

validation message

* Required field.

* Field Type1

* Field

* Field

Field □

Comments

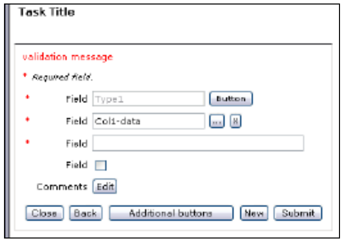
Click to select.

Selected Field

Column1	Column2
Col1-data	Col2-data
Col1-data	abc defg1 ab cdef2
Col1-data	Description - T3
Col1-data	Description - T4

View 2

When a user has searched and/or selected a line item from the table, the screen changes back to the original view, with that form field populated.



Task Title

validation message

* Required field.

* Field Type1

* Field Col1-data

* Field

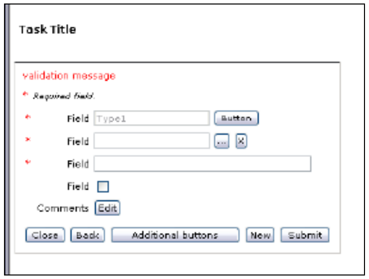
Field □

Comments

View 3

Secondary Flow

Also starts with the standard starting point.



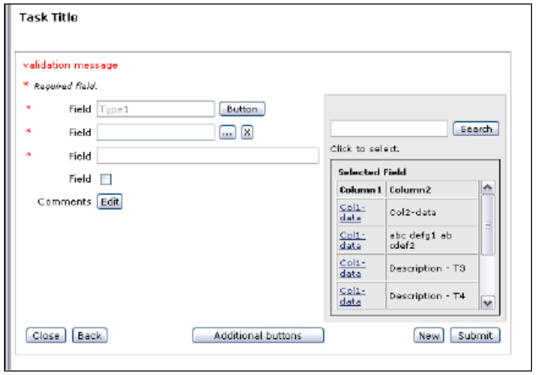
Task Title

validation message

* Required field.

View 1

Clicking the "…" button triggers the screen display change.



Task Title

validation message

* Required field.

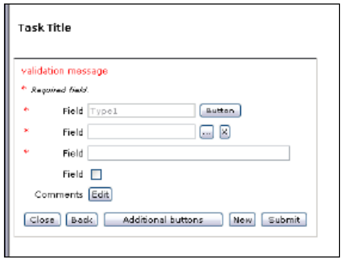
Click to select.

Selected Field

Column1	Column2
Col1-data	Col2-data
Col1-data	abc defg1 ab cdef2
Col1-data	Description - T3
Col1-data	Description - T4

View 2

Clicking "…" again returns the user to the original view. In essence, the "…" button acts as a show-hide toggle. HOWEVER, in this flow, the user has triggered the table to display, and potentially done a search, but has not selected a line item. In this case, clicking "…" a third time will redisplay the table in the same state it was before it was hidden. In other words, the "…" button is simply hiding that aspect of the screen until the user chooses to show it again. If a user completes the selection process (such as in the primary flow), that page element (the search and table) is returned to its default view.



Task Title

validation message

* Required field.

* Field Type1

* Field

* Field

Field □

Comments

View 1

Exceptions

When there are multiple look-ups in a form that are triggered through the ellipsis button, only one should be active at any given time. Clicking the ellipsis button associated to another look-up automatically closes the first look-up and replaces it with the second. The first look-up is then returned to its starting state (in other words, any open searches are dropped when the first look-up is closed).

In some use cases, there is a primary look-up that is not triggered by an ellipsis button. This search/select is never removed from view. Instead, any look-ups that are triggered should appear above the primary look-up. In all other ways, the look-up(s) act as described above. See example, below.

Cursor Placement

For any given screen, the use case's initial task dictates which field or form element will gain focus. Often this is the first form element on the page, but not necessarily so. In most cases this will be a simple matter of placing the user's cursor in the applicable enabled text field. Determination of the appropriate form element must be made on a case-by-case basis, and ultimately is shaped by anticipation of the user's next probable action.

A form element that prompts the user to take action, and then populates the remaining form elements, for example, is the key to the current task. It therefore should gain focus. This standard should be applied throughout any given process flow to keep the task moving forward unimpeded.

Go/Select Button

Go/Select Button	Updates screen/view based on item selection in combination box. Under usual circumstances, this button's enablement is independent of whether a selection has occurred or not. It is occasionally disabled in situations where all or a portion of a form (including the "Go" or "Select" button) are disabled awaiting a selection higher up in the form hierarchy. An example of such a case is SCR22.02.004 Add-Edit Derivative Interest Rate. The entire form is dependent upon choosing a derivative rate code from the right-hand look-up table, and the "Select" button happens to be one of those form elements. This is, however, unusual.

Presentation layer should be controlled through CSS rather than HTML markup. This makes pages smaller, faster, and easier to update.

Never use "Under Construction" or similar phrasing/graphics. If the page is not ready, don't link to it. Only publish when it is constructed.

Do not use underlined text for emphasis, because it is easily confused with a hyperlink; use italics instead, since most browsers by default indicate hyperlinks by underlining.

Blinking text is strictly forbidden.

Flash and GIF animation, if absolutely necessary, are limited to two per page maximum. Only one per page is much preferred.

Web-optimize images for faster downloading using Photoshop/ImageReady or similar specialized software. Smaller images load faster. Image file sizes must not exceed 40 KB. If an image file's size is too large, thumbnails are recommended.

Excluding backend processing time, a page should as a general rule load in under five seconds.

Graphics

Formats

Be aware of the limitations and benefits of the various standard Web image formats. The image format one chooses should reflect the specific function, capabilities, bandwidth limitations and other considerations associated with the graphic in question. Use Photoshop/ImageReady or other specialized software to Web-optimize images (change color depth, dimensions, compression level, etc.). The resulting smaller images will load faster. Image file sizes must not exceed 40 KB. Use thumbnails for larger images.

• *GIF*

GIF (Graphics Interchange Format), created by CompuServe in the earlier years of the Internet, has long been the most popular Web format. Its economical file size makes it ideal for small images such as icons, simple diagrams and other illustrations with large areas of a single color.

GIFs are lossless, meaning they do not degrade (blur or get "messy") as a nasty by-product of too-aggressive compression. Their LZW algorithm reduces file sizes efficiently by finding repeated patterns of pixels, but without sabotaging image quality.

GIFs are limited to a 256-color palette. This makes them a decidedly poor choice for most photographs. GIFs use a fairly crude process called dithering to fool the eye into seeing a third color. GIFs also support single-bit transparency (meaning a single color can be chosen to represent "transparent" in the image), animation and interlacing.

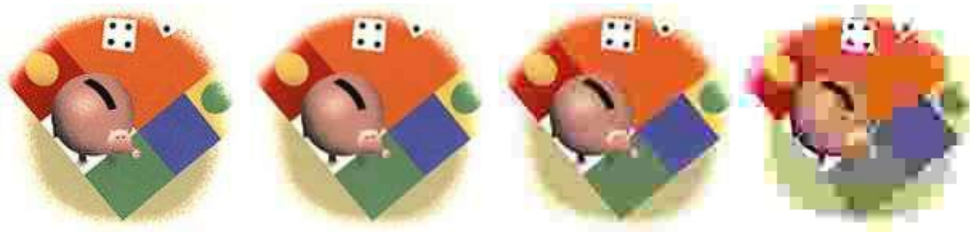
• *JPG*

JPEG (Joint Photographic Experts Group) is a 16-bit, non-dithered format capable of displaying millions of colors simultaneously. Like GIFs, JPGs can be interlaced; however, they lack the animation and transparency abilities of GIFs. They are best-suited for complex images, such as photographs, whereas GIFs are the superior choice for simple graphics with large blocks of color.

JPGs offer compression on a sliding scale: the higher the number, the higher the compression level. That means a smaller file size at the expense of image quality. For example, 0 percent compression offers a clear image with a large file size, whereas 80 percent compression shrinks the file size considerably, but the image quality is noticeably degraded. In principle, 60 percent compression is a good balance between file size and image quality.

Notice compression artifacts creeping into the following graphics. The degrading or “lossy” effects are more dramatic with the increase in file compression settings.

The first image is no compression; the others are 60 percent, 80 percent and 99 percent, respectively.



• *PNG*

PNG (Portable Network Graphic) was invented specifically for the Web in response to certain licensing issues surrounding use of the popular GIF format.

Just as GIFs, PNG-8 (meaning 8-bit color) is limited to 256 colors and supports 1-bit transparency. However, file sizes tend to be significantly smaller, since PNGs save color data more efficiently.

PNG-24 (meaning 24-bit color) is analogous to JPG in terms of numbers of colors. However, PNG-24 is a “lossless” format, meaning that it doesn’t sacrifice quality for file size in its compression algorithm. For this reason, file sizes tend to be larger than their JPG counterparts.

PNG’s claim to fame, however, is alpha channel support. Instead of the rudimentary on/off transparency options of other formats (a given pixel is either transparent or it’s not), an alpha channel specifies a 255-level gradated range of opacity (for example, 0 is fully transparent and 255 is fully opaque). This allows the same graphic to be placed on varying backgrounds and still look good, with the background showing through the pixels that are not opaque.

As for animation, GIFs are preferred at this point for browser compatibility issues. This is likely to change in the future.

- ***BMP***

Microsoft introduced the BMP file (sometimes also DIB, or Device Independent Bitmap) in 1990 as the Windows operating system’s native image standard. BMPs are capable of displaying 1-bit (black/white), 4-bit (16 values), 8-bit (256 values) and 24-bit (16.7 million values) color.

Bitmaps, though viewable by virtually every available browser, are generally considered to be particularly poorly suited for Web use. They do not support transparency of any kind, cannot by themselves be animated, and almost always lack (run-length or RLE) compression to keep their sizes in check, which are significantly larger than their GIF/JPG/PNG brethren. It is best to avoid BMPs altogether for these reasons.

Animation

Animation of any kind (GIF, Flash, etc.) is currently out of scope for all Falcon applications. Limit Flash and GIF animation to two per page, preferrably one.

If multiple search results are returned, the system will allow the user to make payments for different loans using a combination of amount paid, principal due, or principal balance fields.

For example, the system will allow the user to enter an amount paid value for the first loan (loan ID # 111111), select the principal due for the second loan (loan ID # 222222) and select the principal balance for the third loan (loan ID # 333333).

On a specific loan, the amount paid, principal due and the principal balance input fields/check boxes are mutually exclusive. For example, for a specific loan (loan ID # 999999) the system will not allow the user to enter a value in the amount paid field and also check the principal due check box or the principal balance check box and vice versa. Only one of the three fields can be selected for a given loan. The last action taken by the user overrides all other actions.

Note: When the user selects the submit or save button the system will use whatever values are in the amount paid fields to perform validations and to insert or update the current payment request regardless of how the amount paid fields were populated.

Add: hyphen use and spelling standards: **Hyphens and numbers**

1. Use a hyphen with compound numbers from twenty-one to ninety-nine.

fifty-one

eighty-nine

thirty-two

sixty-five

eighty-one

2. In written fractions place a hyphen between the numerator and denominator.

two-fifths

one-third

three-tenth

nine-hundredth

[*Exception*] if there is already a hyphen in either the numerator or the denominator, you omit the hyphen between the numerator and denominator.

sixty-nine eighty-ninths (*not* 'sixty-nine-eighty-ninths')

twenty-two thirty-thirds

3. Use a hyphen when the number forms part of an adjectival compund:

France has a 35-hour working week.

He won the 100-metre sprint.

Charles Dickens was a great nineteenth-century novelist.

Web Writing Tips

Brevity

Be succinct and write for scannability. For instance, write half the amount of the text you would for hardcopy publication, and avoid long, continuous blocks of text. As a general rule, paragraphs should be no longer than three sentences. Seventy-nine percent of Web users scan pages; they do not read word-by-word. Highlight liberally (about three times as often as in print) to make keywords stand out.

Simplicity

Use simple sentence structures with an informal voice. As it is, people read about 25 percent slower when reading from a display. Use bulleted and numbered lists to make information delivery more efficient in a smaller package.

Top Placement

Put the most important information at the top. Use an inverted pyramid approach, such that the information becomes increasingly more detailed as one reads.

Hypertext

Make use of hyperlinking to split up long information into multiple pages. Link to supplemental information like definitions, abbreviations, reference information and background reading, should the user wish to delve deeper into a particular subject.