

**State of Florida**

**Department of Environmental Protection**

JEE User Interface Guidelines

**GDE-09061802.1.0**

|  |  |
| --- | --- |
| **Date:** | 6/18/09 |
| **Version:** | FINAL 1.0 |
| **Authored by:** | Office of Technolgy and Information Services |

Table of Contents

Section Page

1. Introduction 4

2. Compliance with DEP’s Web Standard 4

3. Single Point of Entry for Departmental Applications 4

4. Consistent Form Layout 4

4.1 Overview 4

4.2 Default Screen Size 4

4.3 Stylesheets 4

4.4 Fonts 4

4.5 Overall Form Layout and Tiles 5

4.5.1 Overall Application Look 5

4.5.2 Header Tile 7

4.5.3 Menu Tile 8

4.5.4 Detail Tile 9

4.5.5 Footer Tile 10

5. Standard Forms / Form Parts 11

5.1 Single Entry Point 11

5.2 Splash Screen 11

5.3 Search Results 11

6. Standard Basic Controls 11

6.1 Text Boxes 11

6.2 Links 11

6.3 Buttons 11

6.4 Radio Buttons 12

6.5 Check Boxes 12

6.6 List Boxes 12

6.7 Dropdown Combo Boxes 12

6.8 Spin Boxes 12

7. Standard Advanced Controls 12

7.1 Menu Control 12

7.2 Dot Bar Control 12

7.3 Tab Control 16

7.4 Data Navigation Control 16

8. Use Java Script 16

9. Data Validation & Messages 17

9.1 OROV & Other Functional Messages 20

10. New Window/Pop-Up Usage 21

11. Resources 21

JEE KickStart 21

# Introduction

This documents specifies the Java Enterprise Edition (JEE) User Interface Guidelines at the Department of Environmental Protection (DEP). Map and Hand-Held based applications are not required to follow these guidelines.

# Compliance with DEP’s Web Standard

All web-based applications must comply with Section 508 accessibility standards at <http://depnet/bisnet/web_author/references/ada.htm>.

# Single Point of Entry for Departmental Applications

All DEP applications, reports and tools must be accessible through a single, web-based access point known as the DEP Portal. The DEP Portal uses Oracle Single Sign On authentication to allow users to access their specific applications. Movement from the DEP Portal to the functional application should appear seamless to the user.

# Consistent Form Layout

## Overview

The application framework is based upon a multi-panel layout including several ‘tiles’ for the various components of the application. The mandatory tiles are Header, Navigation, Body, and Footer. Information/Control tiles are optional. All tiles should be implemented in frames containing Struts Tiles, Java Server Pages, Java Server Faces/Facelets, or other technology as appropriate.

## Default Screen Size

Applications should be designed for 1024 by 768 screen dimensions. This does not preclude “vertical scrolling” as required by the application functionality, or horizontal scrolling of certain types of metadata-driven tools.

## Stylesheets

All DEP web-based applications shall use the layout and graphics specified by the Office of Communications and shared in the master STYLES.CSS as provided by OTIS in the resources listed below. Each tile will use the departmental standard style sheet which includes entries for heading, navigation, and buttons.

## Fonts

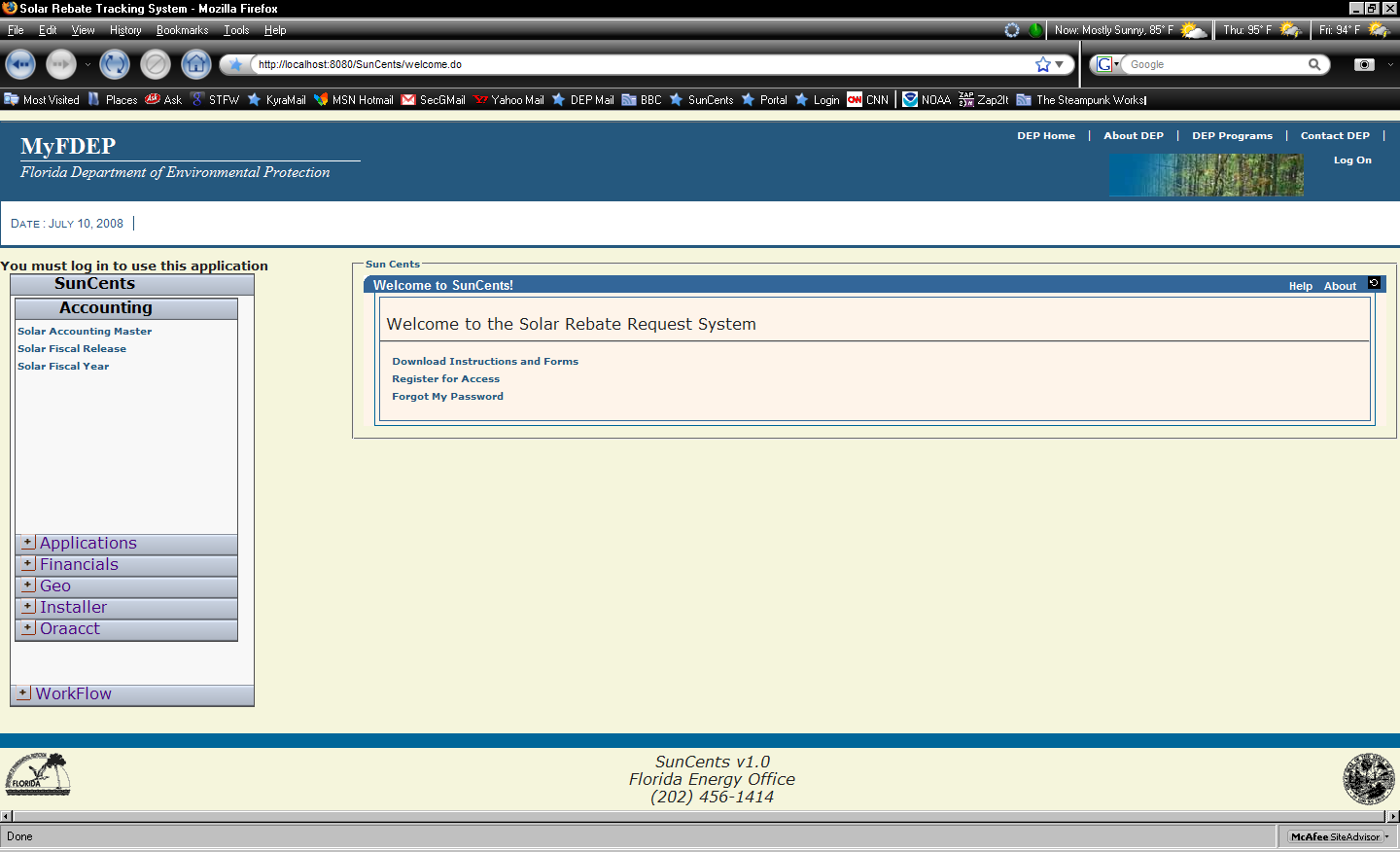
The fonts are defined in the STYLES.CSS provided in the JEE Kickstart.

## Overall Form Layout and Tiles

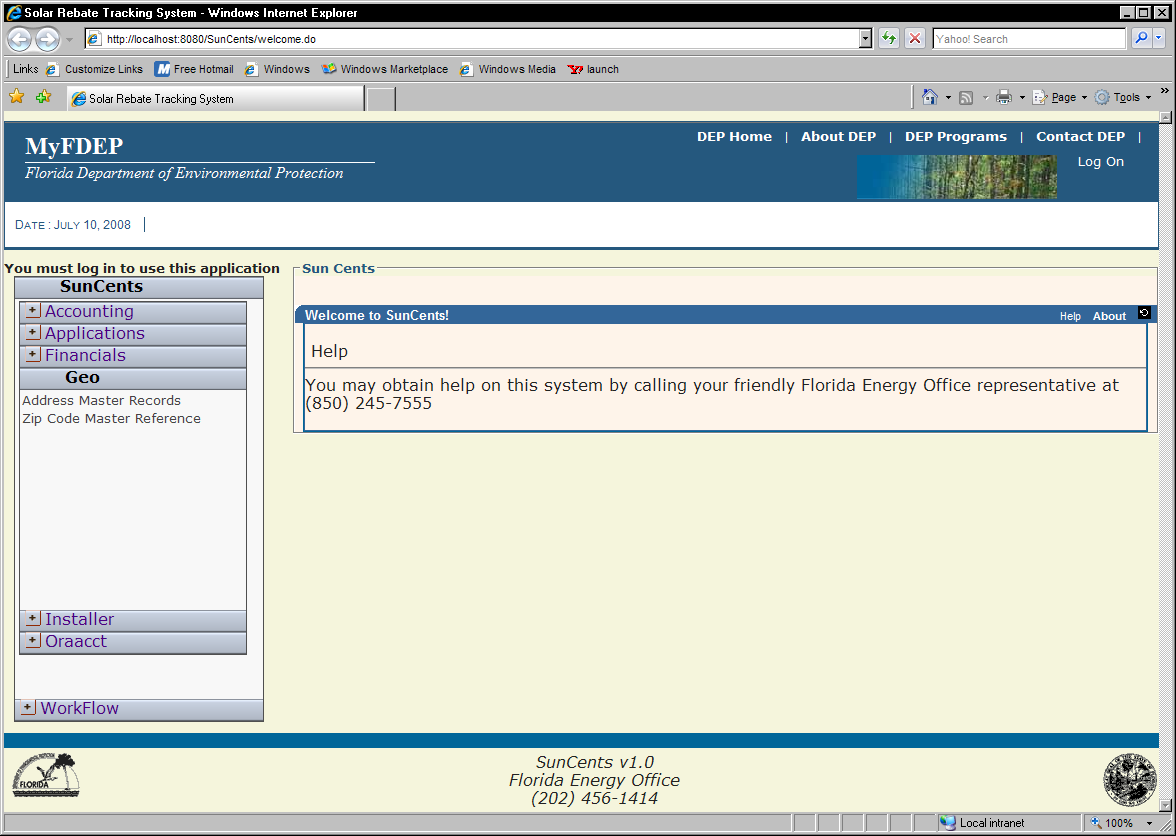
### Overall Application Look

The following three figures provide examples of the tiles-based layout for a standard page that shows the three primary panels required for each page. These are the “normal” mode, an associated Help text, and an About panel. These are implemented as hidden “tab panels” and the Help/About buttons are used to hide/show those panels. There is also a “Refresh” icon which can be used to return to the main panel, and the panel name is a link to itself as well. The application name in the Legend box is a link to the home page of the application.

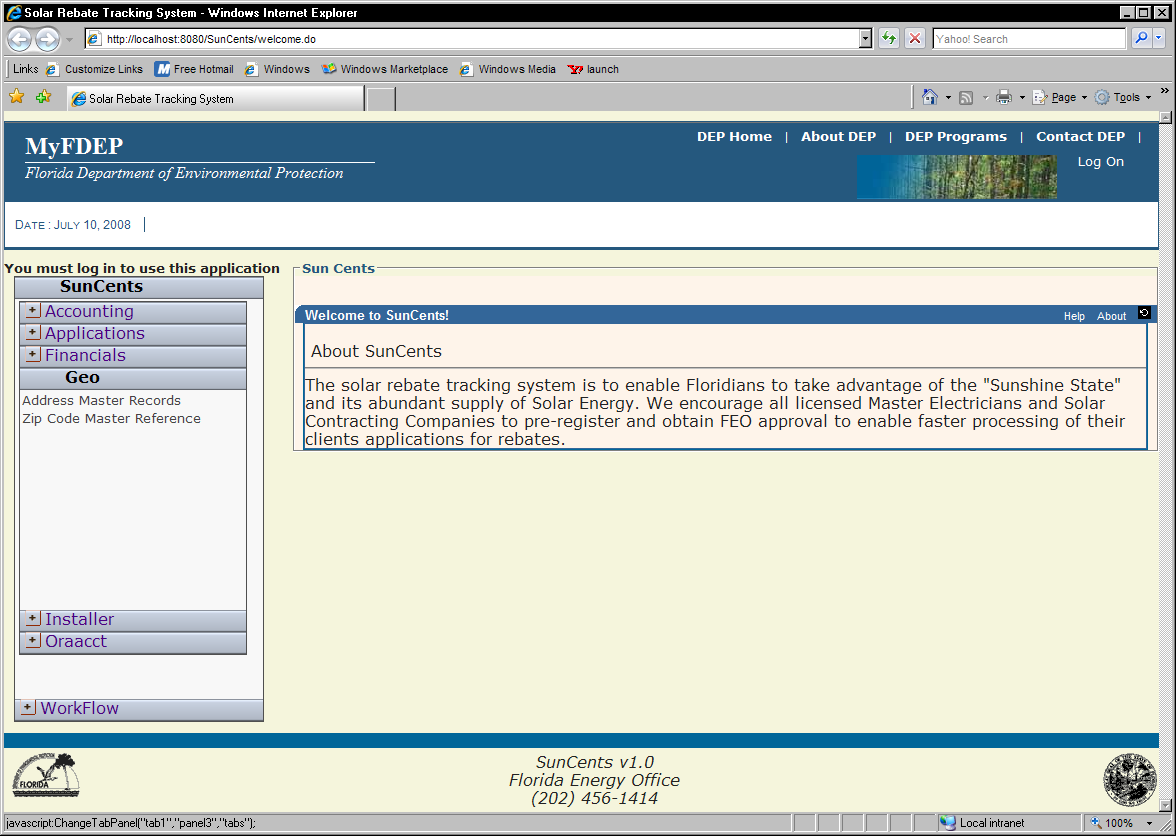
**Panel 1, the primary display page:**



**Panel 2: An example of a “Help” tab:**

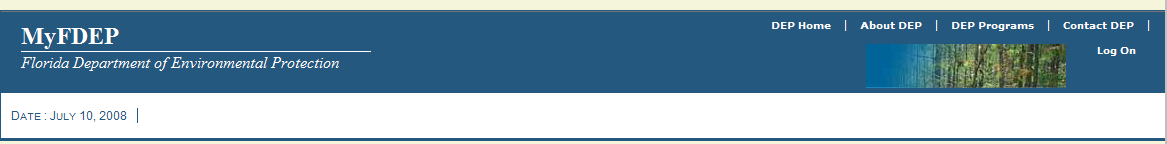


Panel 3: “About” tab



### Header Tile

The top line of the Header Tile must contain links to *DEP Home*, *About*, *Programs*, and *Contact DEP*. The following figure provides an example of a typical header:



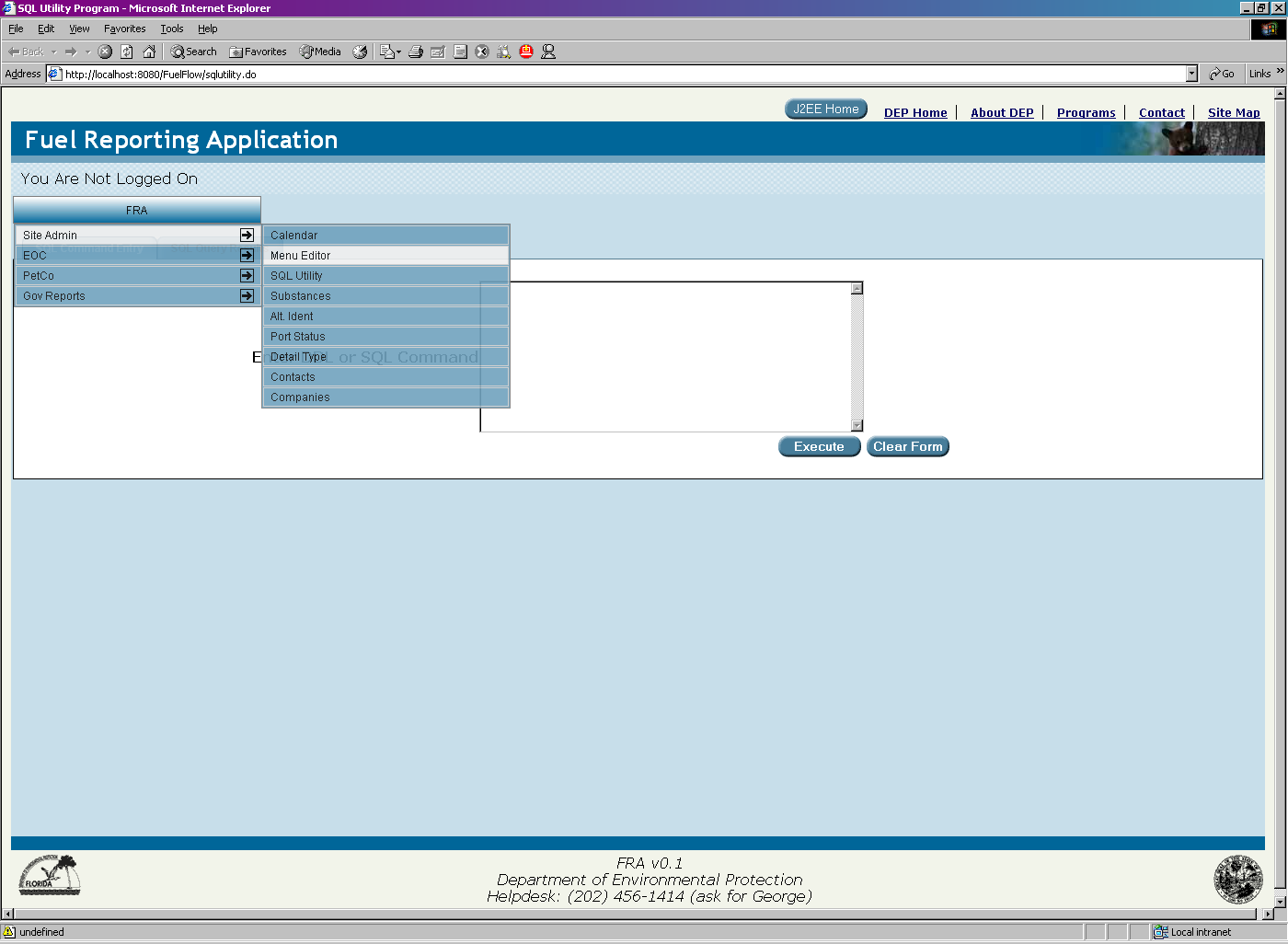
When using the Single-Sign On, the user’s name will appear as: “Welcome *User Name*”

### Menu Tile

#### Horizontal Option

The content of the Horizontal Style Navigation tile will be one or more application specific menu options displayed on a horizontal line, implemented as vertical “fly” opening, utilizing the “hand” cursor over the menu items to indicate to the user the “link” condition. This menu style is recommended for usage with relatively fewer items in the menu, and for those pages which need full-width for data.

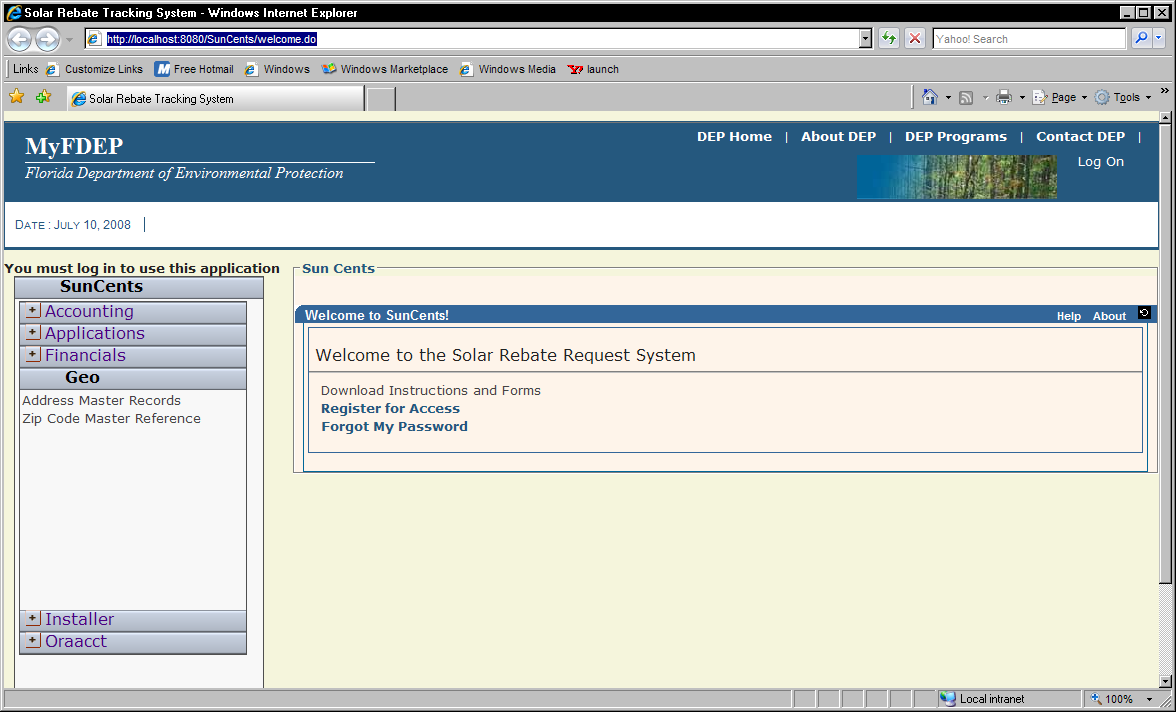
The following figures displays an example of a horizontal navigation tile. Note the *Execute* and *Clear Form* buttons; all buttons must conform to this standard and use these images:



#### Vertical Option

The content of the Vertical Style Navigation tile will be one or more application specific menu panels, implemented as a vertical opening “Accordion” based menu The content and usage of Styles will be standardized and implemented in the common.css which is to be jointly developed by the IT Coordinators and their representatives. This .css file will include the hover effects colors for menu highlighting as well as the cursor change to “hand” style for links. The vertical style is recommended for applications having a large number of navigation links.

An example of a Vertical (left-hand side) navigation tiles is shown in the following figure:



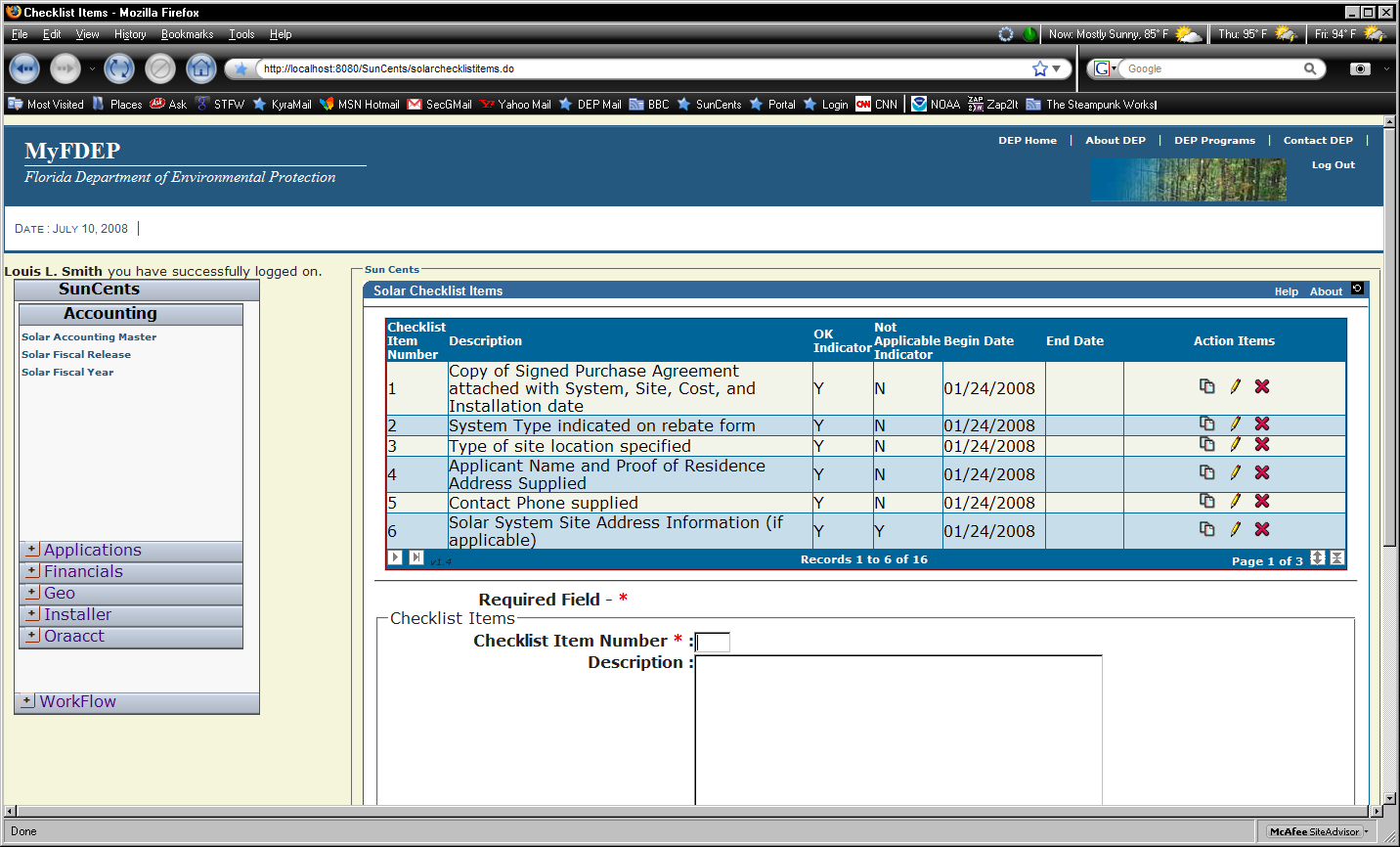
### Detail Tile

The Detail Frame tile must be the central portion of the screen and be implemented with a standard nested table structure. Buttons must use the image as specified in the figure in section 4.5.3.1.

If a Data Frame contains a listing type display, a button based navigation bar must be placed on the frame just above the bottom command bar. Data Panels must be implemented as Tab Panels when large amounts of data are required.

The SAVE button may be “grayed out” until the user has changed a data entry field on the screen, at which time it will be reset to “active”.

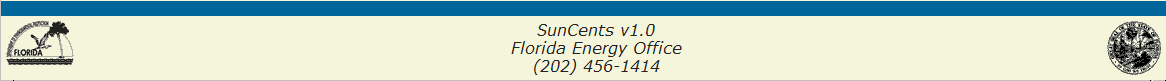
If a Data Frame contains a long listing of entries, the “table” must be implemented with highlighting and optional column sorting. If the number of records exceeds the users preference for viewing (set in User Preferences file), then a record navigation bar must be placed on the screen above the command buttons. The following figure shows an example of a navigation bar:



The use of a Struts or JSF standard pagination taglib is recommended.

### Footer Tile

The footer tile must be 3 lines of text, centered on the screen. These three lines consist of the Short Project Name and Version, Department and Division, and Helpdesk Support #. Additionally, the DEP logo must be on the left side and the State Seal must be on the right side of the footer tile. The following figure provides an illustration of a footer for the SunCents project:



# Standard Forms / Form Parts

## Single Entry Point

A single URL must be used for all DEP JEE applications. This URL is the primary Portal screen with Portlets whose content is to be determined by OTIS.

## Splash Screen

Each application may have its own “splash” screen as shown in the “Welcome” page example above. This may be context sensitive and subject to edit by a site administrator as required by the applications user requirements. If a site does not have this “initial message” requirement, then the most commonly used data entry screen may be its entry page.

## Search Results

Search results pages must limit the number of records returned and provide for pagination of the result set using standard functions of the framework selected for the project, based upon each projects/functions search requirements. The search results may be displayed in a Grid, on a report, or in a data entry form if only 1 record is returned. This is dependent upon the functional requirements of the search. Complex searches may be developed on a Tabbed form, with the search on the first tab, and the results on the second, as in the example pages showing the SQL Utility.

# Standard Basic Controls

## Text Boxes

Text boxes are to be sized appropriately for the contents, so as to avoid horizontal scrolling of data whenever possible. Larger text areas may be entered via a TEXTAREA as opposed to a singular text box. As all data elements have a length attribute, this should be used to set the size of the text box as well as the maximum size allowed.

## Links

All links are to use the style included for text-decoration of underline while hovering

## Buttons

Command buttons on screens should be context sensitive for usability. As an example, when in Insert mode, the button should be labeled “Insert”. When in Edit mode, the button should be labeled “Update”. Ambiguous labels must be avoided. All buttons will use the supplied button image from communications.

## Radio Buttons

Radio buttons must have one button selected as the default upon display of a “blank” entry screen. Based upon the requisite labeling text, the button layout should be horizontal if possible rather than a vertical box. If the requisite labels are long descriptive text, then a vertical layout is preferred.

## Check Boxes

Based upon the users data requirements and size of the requisite box labels, the Check Boxes may be either horizontally or vertically aligned the same as Radio Buttons. All “default” boxes should be checked upon display of a “blank” entry screen, per user requirements.

## List Boxes

The usage of List boxes based upon reference tables is strongly encouraged in lieu of raw data entry of common data. The “view window” sizing should be based upon whether the box is single or multi-selectable per user requirements.

## Dropdown Combo Boxes

The usage of programmed dropdown/combo boxes is dependent upon user requirements and may be implemented using standard tools.

## Spin Boxes

The usage of Spin Boxes is encouraged for selection of range based numeric entry where appropriate to the users need.

# Standard Advanced Controls

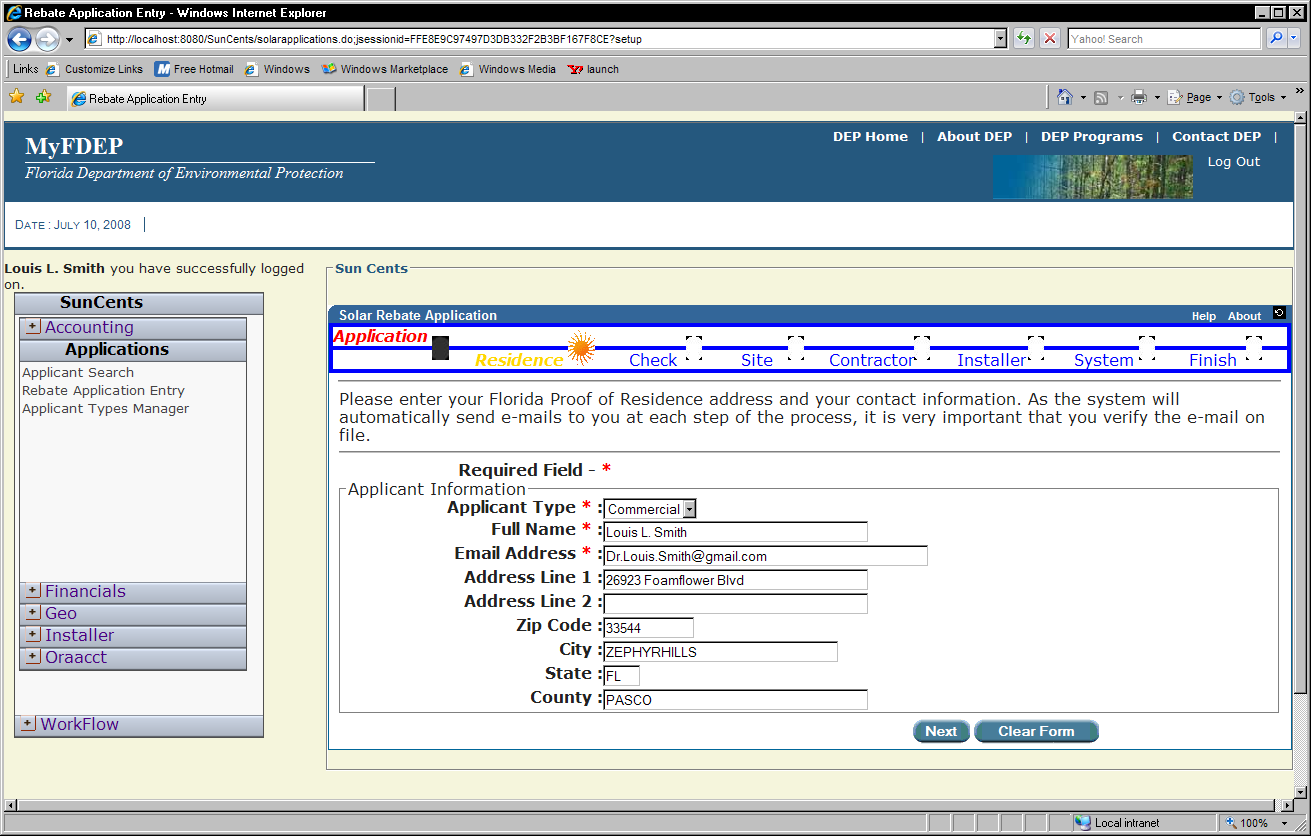
## Menu Control

The menu panels will utilize the standard as supplied by OTIS and will be security role based.

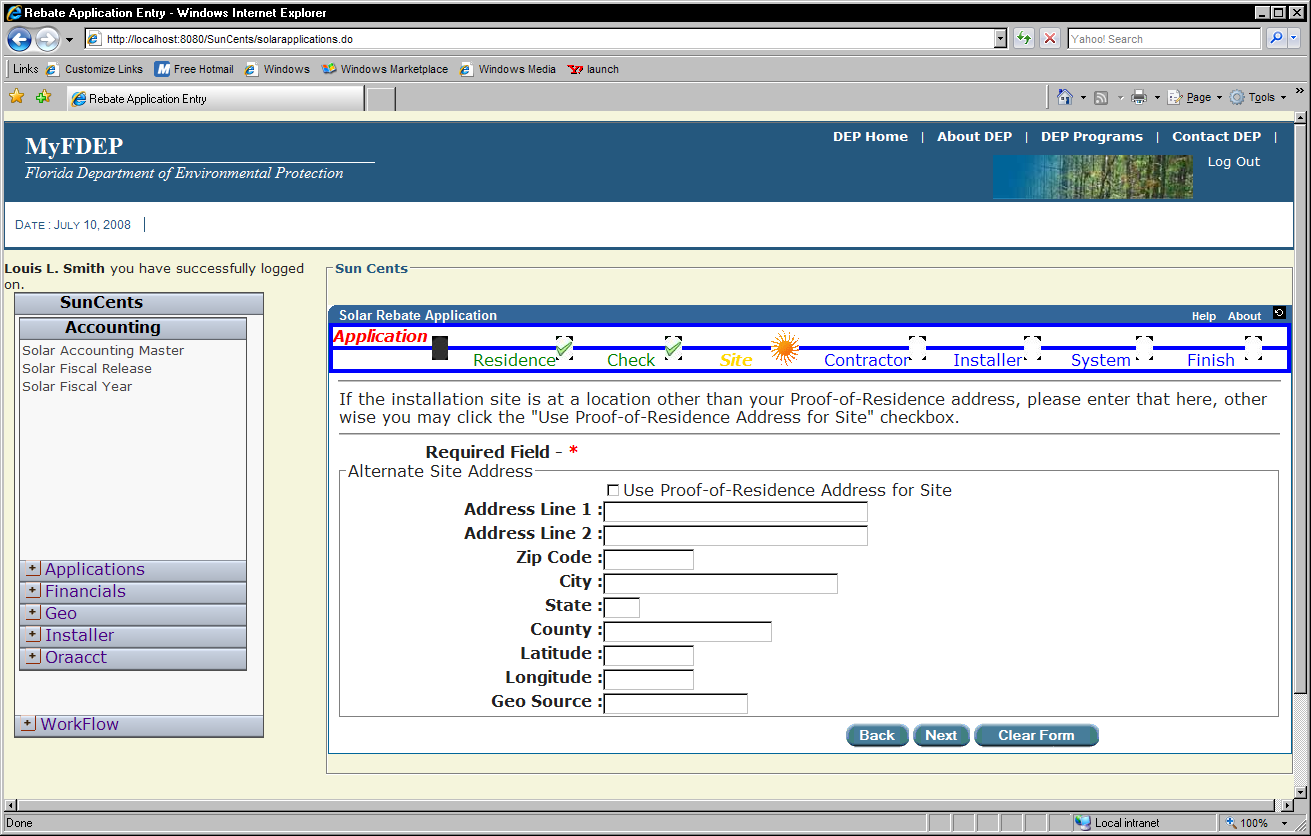
## Dot Bar Control

A dot-bar controlled page should be used for complex forms which require the user to supply a larger amount of information across several steps or logical groupings of data.

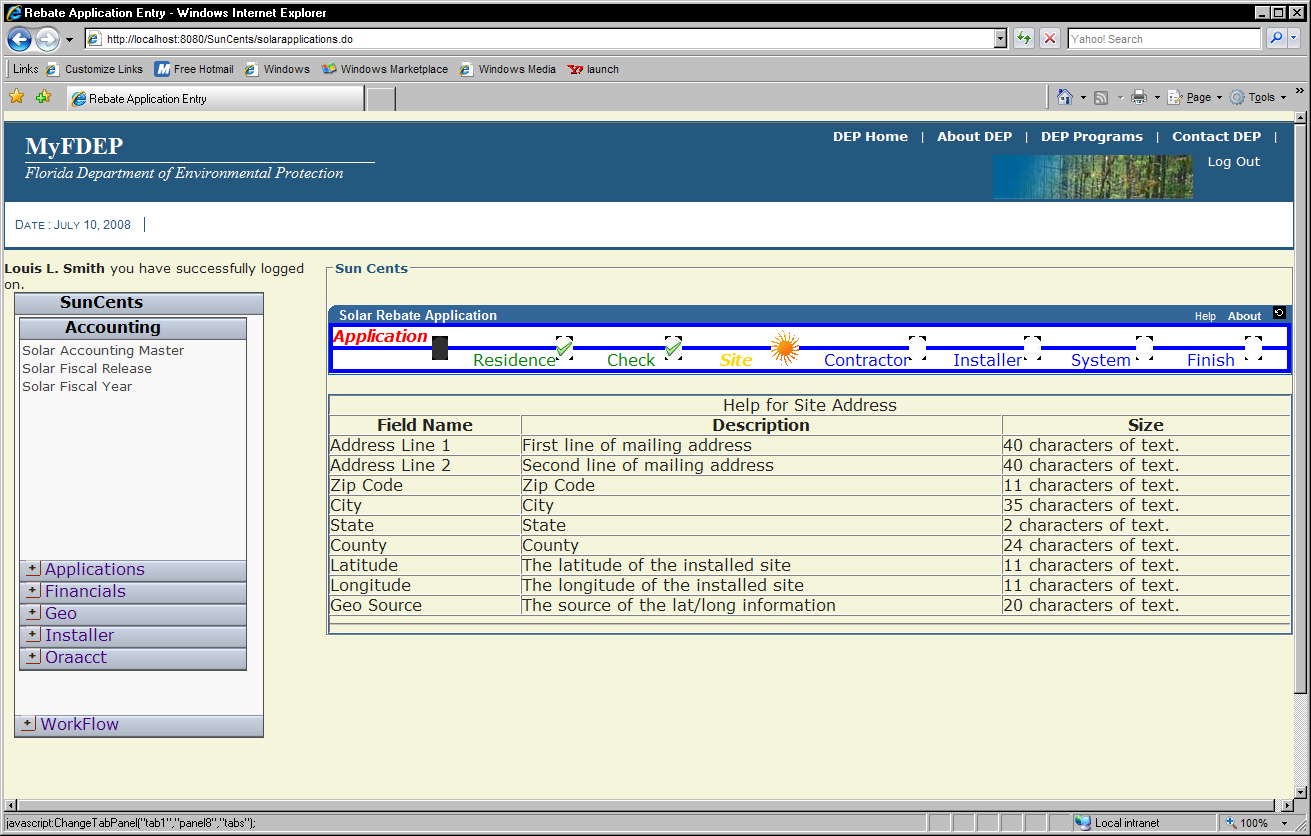
This is an example of a data entry form utilizing the OTIS dot-bar manager control with customized images for the Solar Rebate System. This first panel shows pre-populating information from the users log-on credentials:



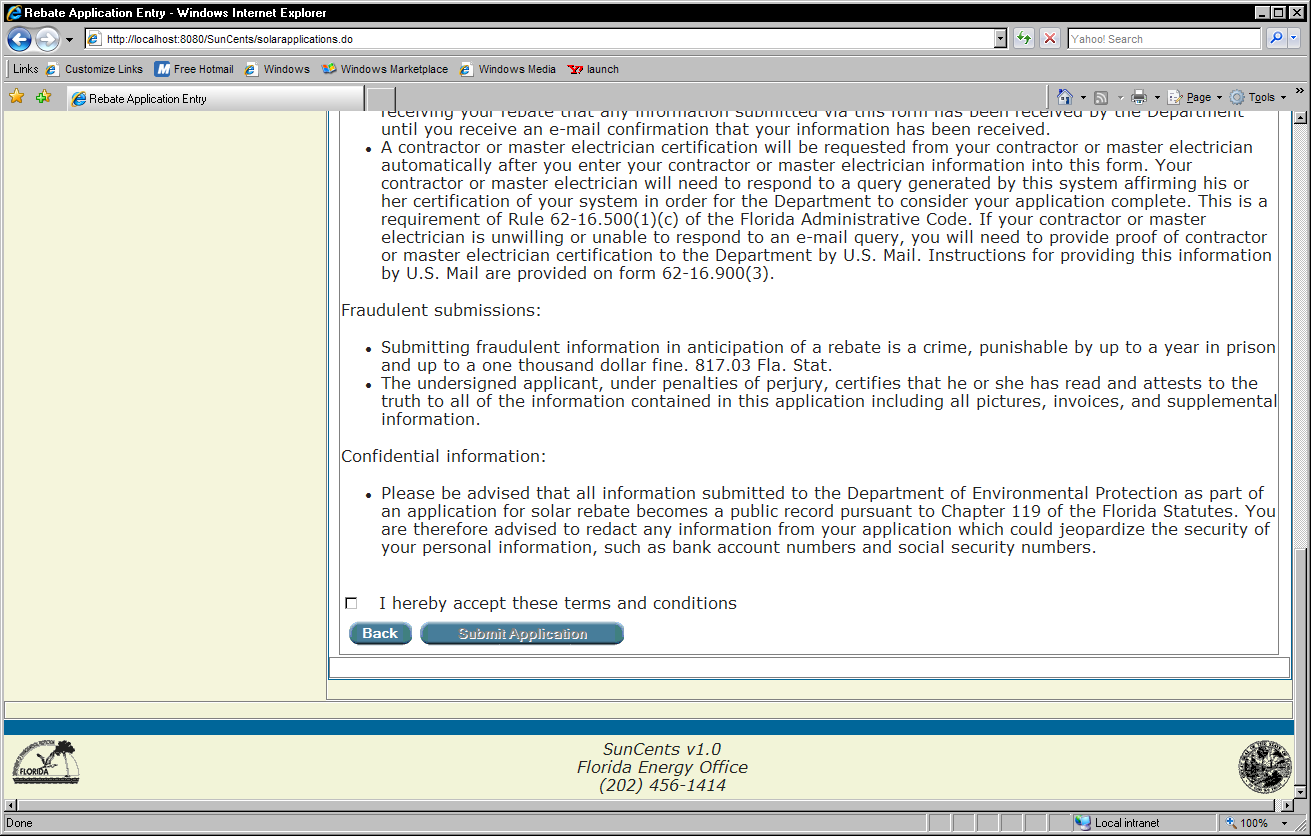
This next screen shows the status at the 3rd step, showing Green Check Marks for successful completion of steps, and the highlighted Sun image for the current step:



This next example shows the dot-bar-step-based Help panel:



All dot-bar based screens will use the BACK and NEXT buttons as appropriate, and will use a final “Finish” screen where the user can review all information submitted and perform the final submission. This page may also include standard legalese about the submission, and utilize the clickable checkbox for “acceptance of terms and conditions” which will enable the final submit button, as seen here:



## Tab Control

Tabbed panels should be used for hierarchical data viewing when possible. The tab images are supplied with the STYLES.CSS and supporting files and are to be used on all pages.

## Data Navigation Control

The usage of data navigation controls is recommended on data viewing screens which have the potential of returning large numbers of records. The usage of a standard pagination control with data navigation is also recommended.

# Use Java Script

Javascript should be implemented in files with a .js extension to allow the browsers to cache them for further usage.

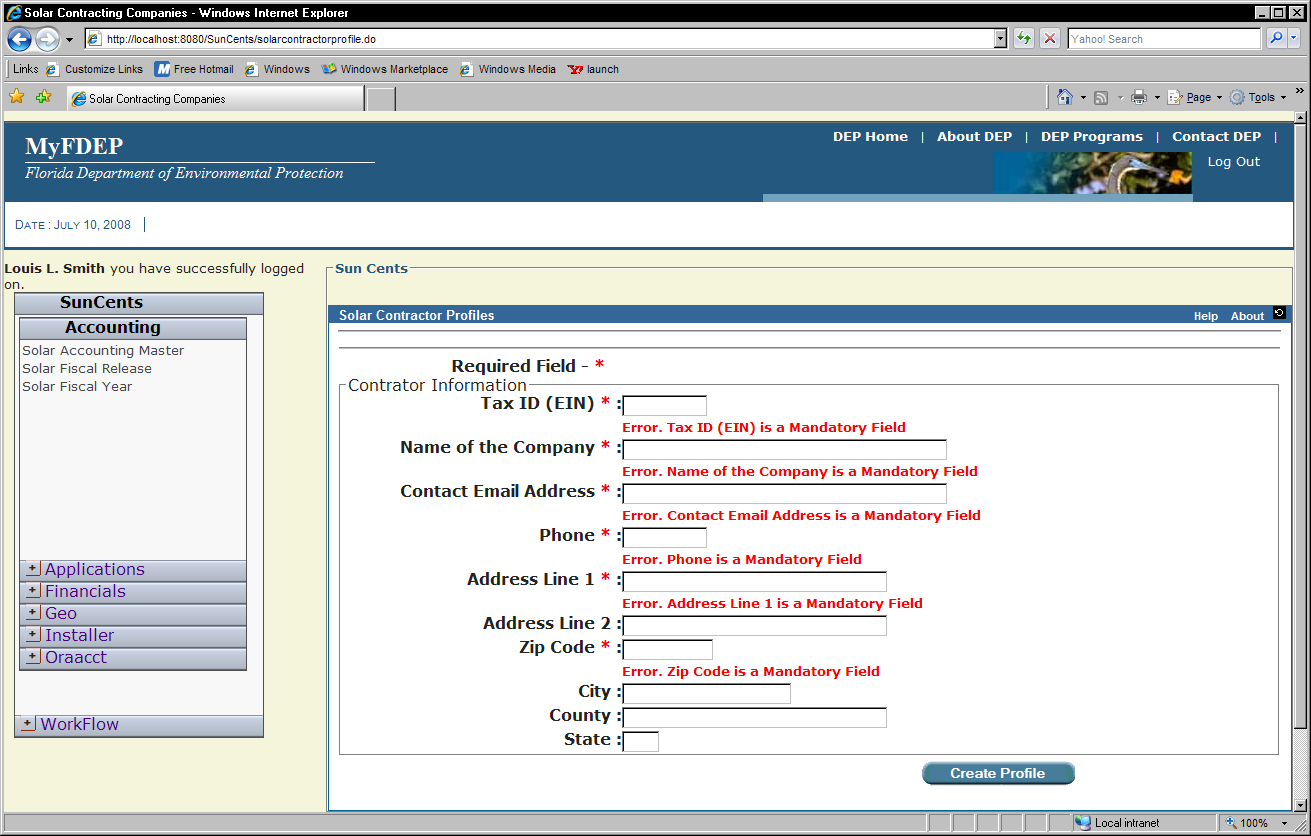
# Data Validation & Messages

Data Validation should be implemented using the Jakarta Commons library routines. These routines should be implemented in the Bean behind the form for first level validation (existence and type of data); using the XML Validation for content (email, numeric range, etc); and application code can also display application specific validation errors.

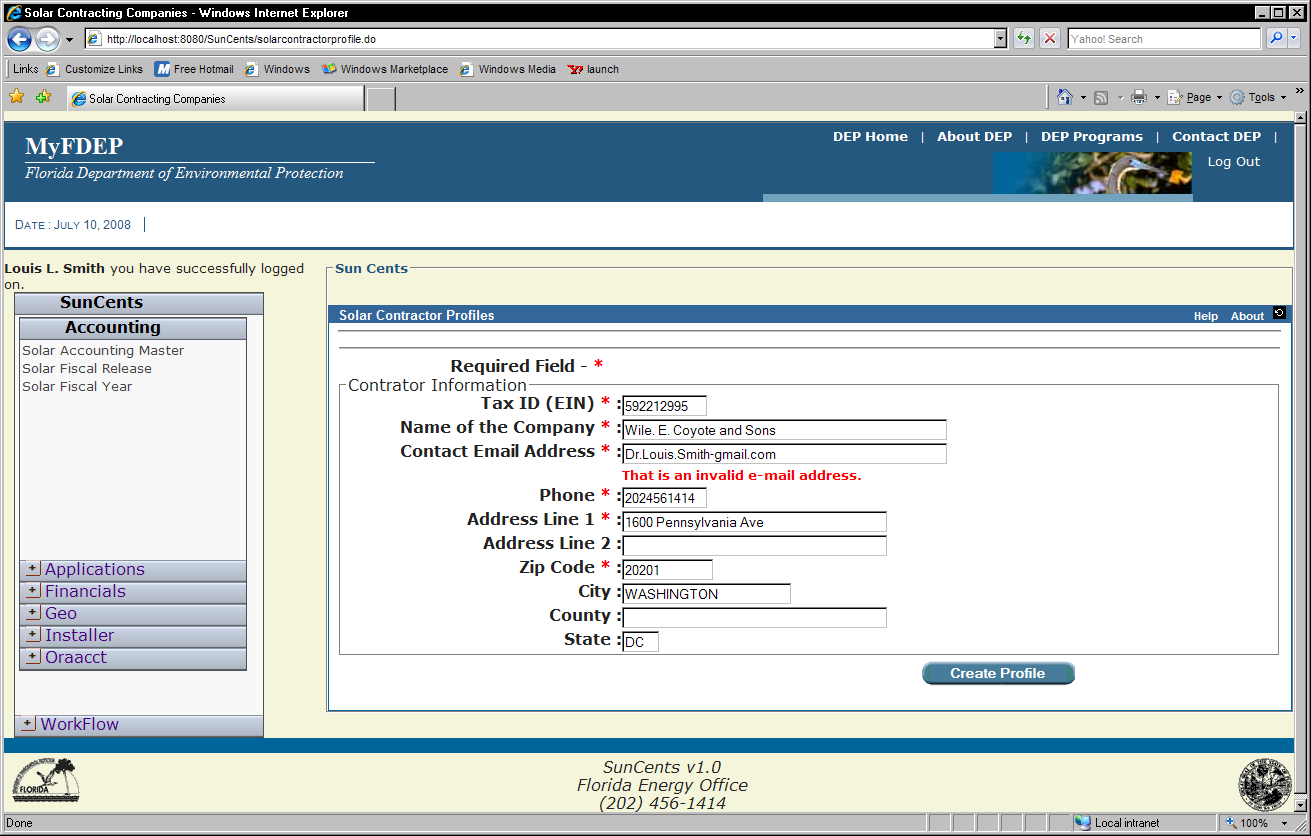
As Data Validation messages are directly tied to data entry fields, they should be displayed with the field.

Examples of the 3 primary types of Data Validation messages are shown in the following figures:

**First Level: Existence Check inside a “Form” or “Backing” Bean in Java using standard Validate method:**

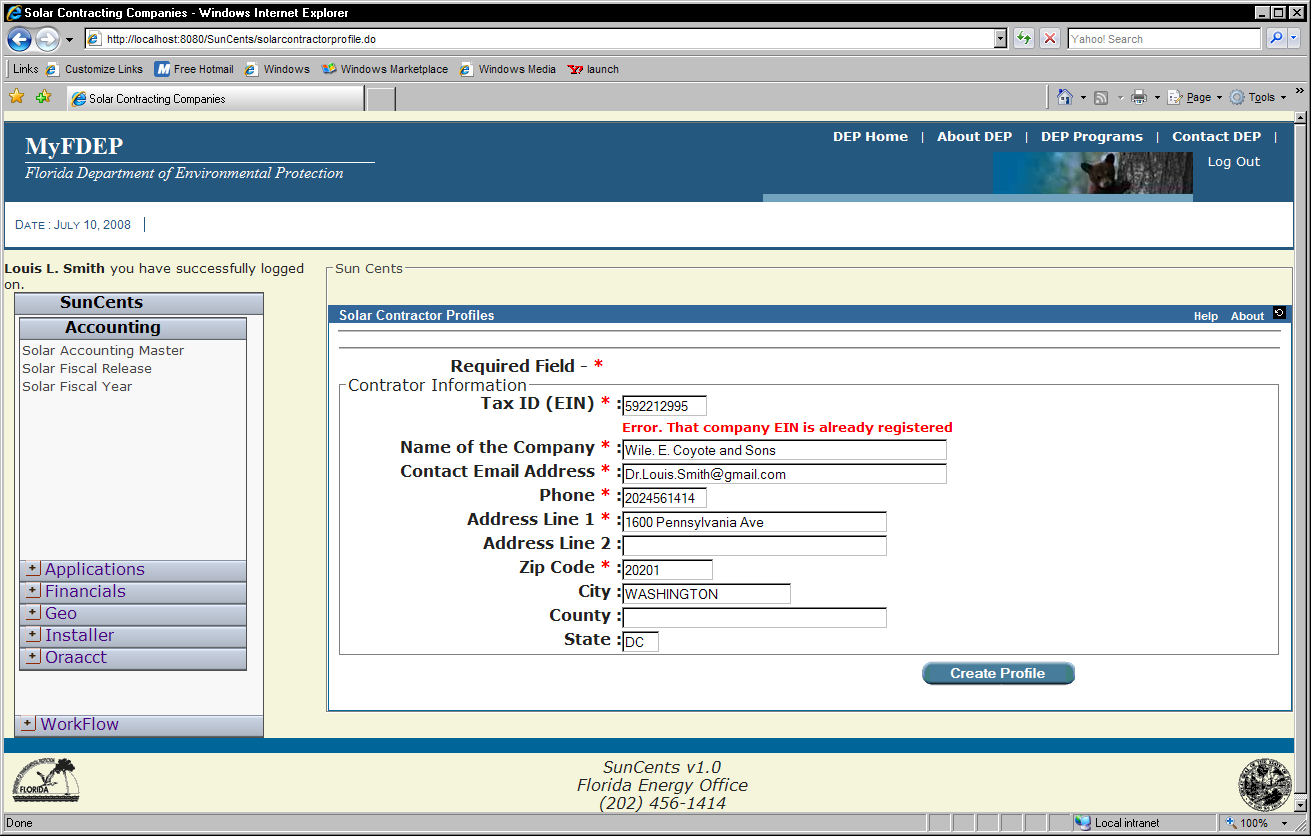


**Second Level: XML Validation Routines for content of fields using the Jakarta Commons library:**



**Third level: Application showing an internal level error.**

This message comes from deep inside the application when it has read the database and determined that a field is causing a problem – here, it shows a user’s email has already registered for access.

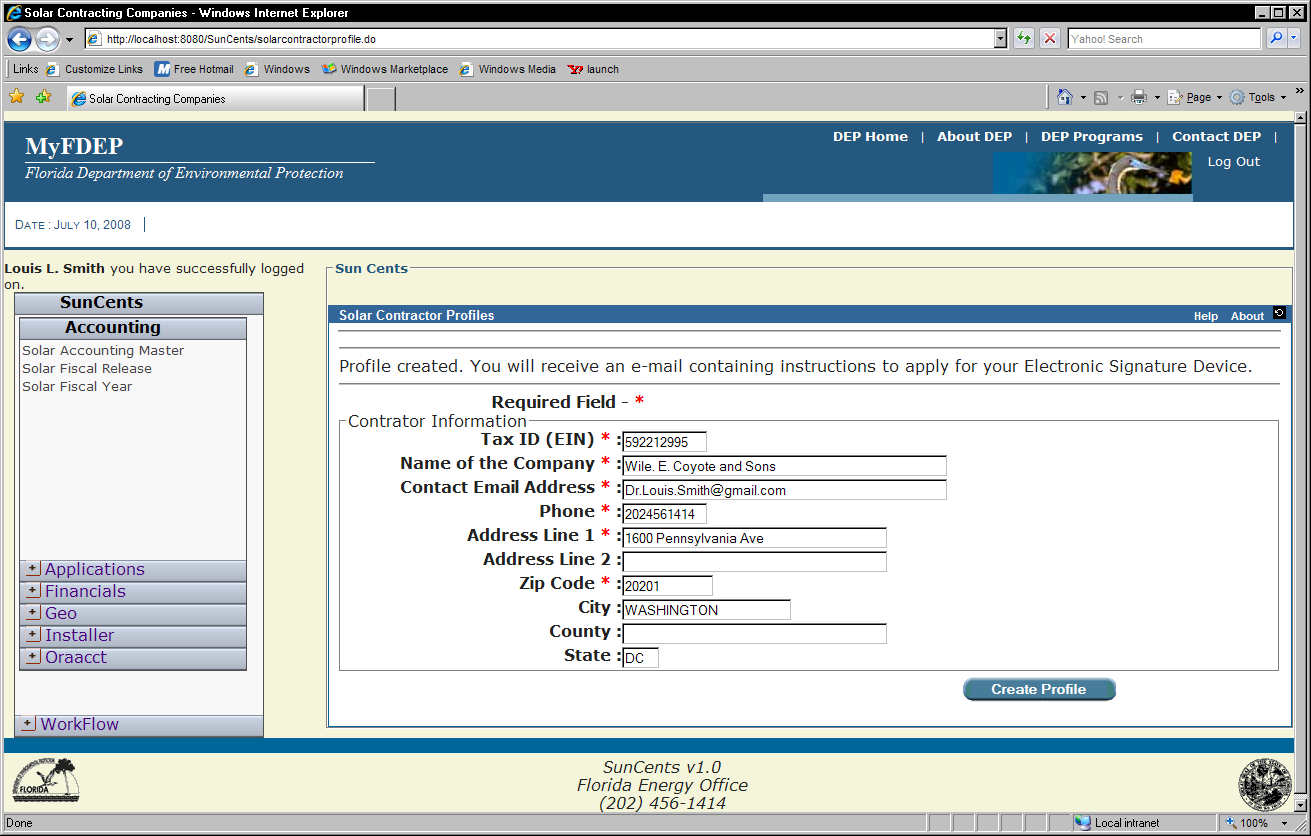


## OROV & Other Functional Messages

As the classic definition of a Use Case is “A Sequence of Actions a System Performs Yielding an **Observable Result Of Value**”, we should display the OROV message on the screen, above the data entry area.

This example shows a message confirming creation of a user profile, as well as the notice that an e-mail has been sent as the OROV from the last button click. This technique can also be used for longer text messages or Functional Errors to the user which are not specific to a field, but to the function overall. Functional Error messages would be narrative descriptions of an applications inability to perform some task.

The following figure shows an example of an OROV:



# New Window/Pop-Up Usage

For applications which require full-width and height of the screen, a “new window” paradigm is encouraged. The new window is not required to use the master layout.

New Windows should also be used for viewing of system documentation (Javadoc) and logs during development. As the menu is role based, only those users flagged as “developer” or “dba” will see these menu options.

For extremely large and complex data entry that can not be re-engineered into multi-tabbed panels a new or pop-up window may also be used.

Pop-Up windows are also encouraged for data entry helpers as shown in this “Date Selector” pop-up window:



# Resources

## **JEE KickStart**

The complete JEE application Kickstart contains the styles, images, scripts, and foundation pages for a new JEE based application. The functionality it includes covers the Struts/Tiles framework, resources, all images, stylesheets, and javascript files as used in the application presented within this document. The currently supported Oracle Single-Sign-On is also included. Upon request to the OTIS Applications Services Manager, Kickstart will be customized for and delivered to each project.

