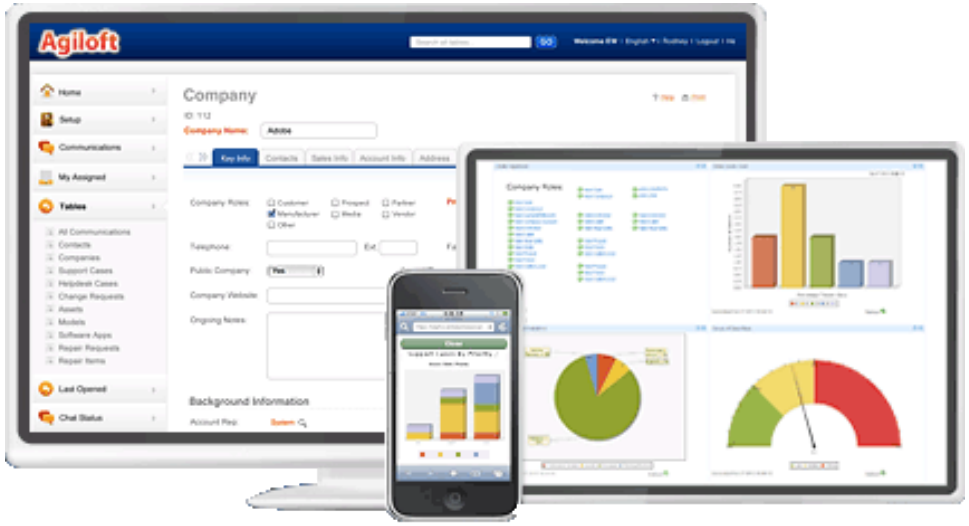
Unit 19: End User Interface



Questions Covered

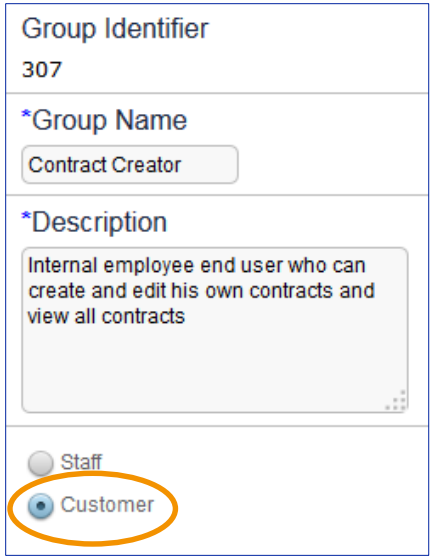
- How does the main **End User Interface (EUI)** work?
- What alternative end user interfaces exist?
- How do we make changes to the EUI?
- Can I add pages to the default setup?
- How do we change the logo and customize branding?
- What determines the tabs each user sees in the EUI?
- Where can I learn more?

Unit Overview

- In this Unit, we will be learning about the default **EUI** setup.
- We will briefly cover the relevant **tags** and HTML code used in the EUI Templates, and the **macros** that help you display tables, images, records and other information through the EUI.
  - If you need a refresher on HTML, please read the preamble to Unit 19, the HTML unit.
- We will learn how to provide end users with access to tables and forms, and begin to customize the interface.

End User Interface Functions

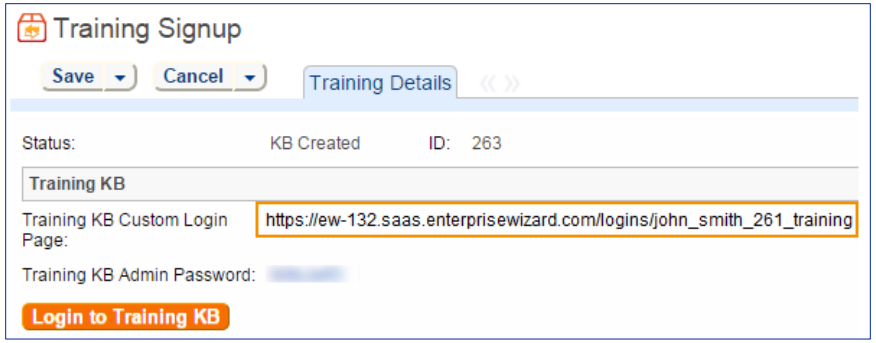
- In Unit 6 we learned that end users belong to groups designated as Customer on the Global tab of the Groups wizard (**Setup > Access > Manage Groups**). They access the system with an end user license through the **End User Interface (EUI)**.
- From the **EUI**, end users can be permitted to:
  - Search, view, and print FAQs.
  - Create records in any table.
  - View records in any table.
  - Edit records they "own" in any table.
  - Run and save simple and complex searches.
  - Edit their user profile.
- End users may be external customers or company employees. In a typical implementation, they are users who submit records or requests but do not work on them.



- The default setup covers a range of common uses such as helpdesk and support applications, as well as contract and document management.

Accessing the EUI

- Let's find the **custom login page** for your Training KB before we continue.
- A link to the Training KB Custom Login Page is available from your training signup record. A link to the training signup was sent to your email address when you initially signed up for administrator training.



- The URL will look similar to: https://ew-132.saas.enterprisewizard.com/logins/john_smith_261_training/

The last string is the name of the KB. It will be in the form firstname_lastname_id_training, based on the information you supplied when signing up for this course.

- Enter the URL in your browser to access the KB's **custom login page**. You may want to bookmark this page for convenience during this unit of training.

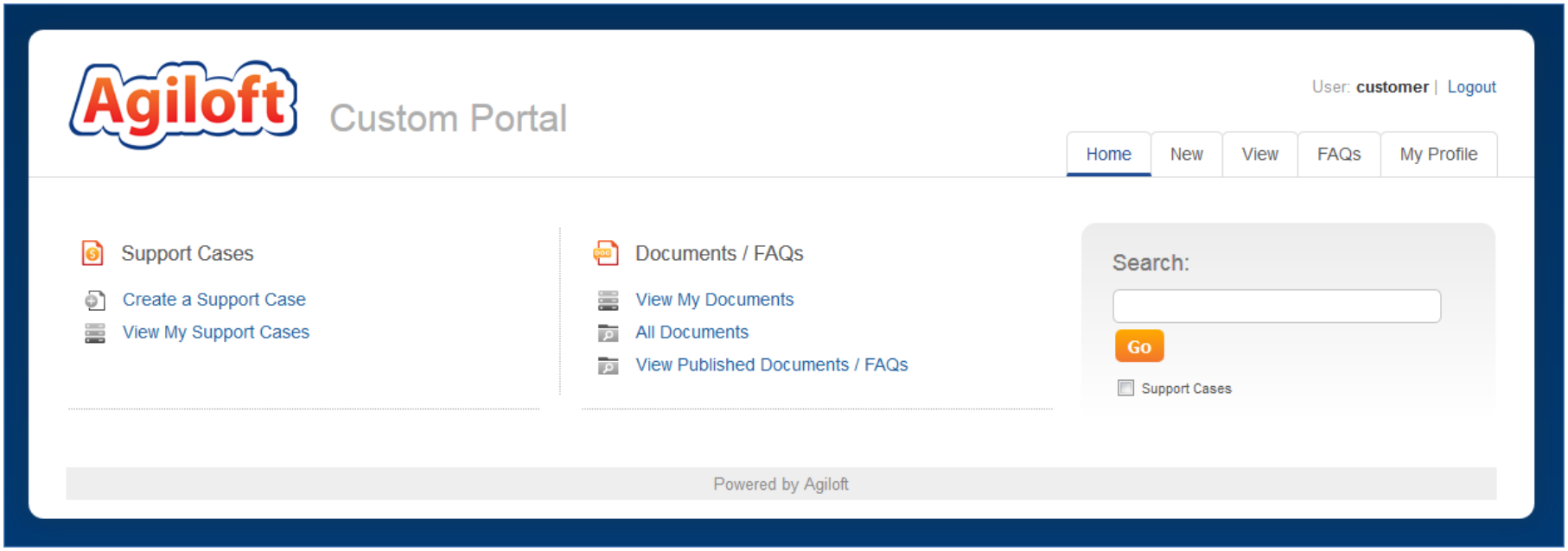
Accessing the EUI (continued)

- Enter customer for the username and password (a default test user with these credentials is included in your Training KB).
- Check the End User radio button and click Log In.
- Once logged in, click on a few of the tabs and links to see the options available to our test user.

- You can log in to Agiloft as multiple users in separate browser tabs. It may help to follow along with our discussion as an admin user in the staff interface and simultaneously view the EUI as a test end user in another browser tab.
- The next slide shows the default EUI landing page.

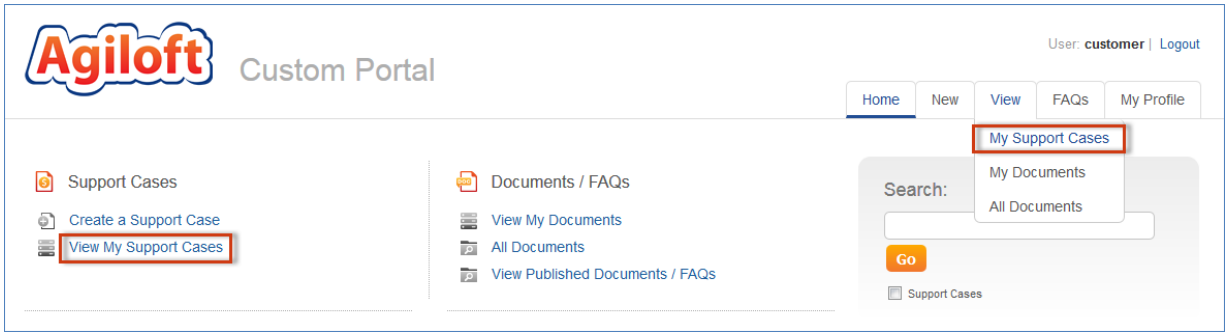
Default End User Interface

- EUI Customer landing page:

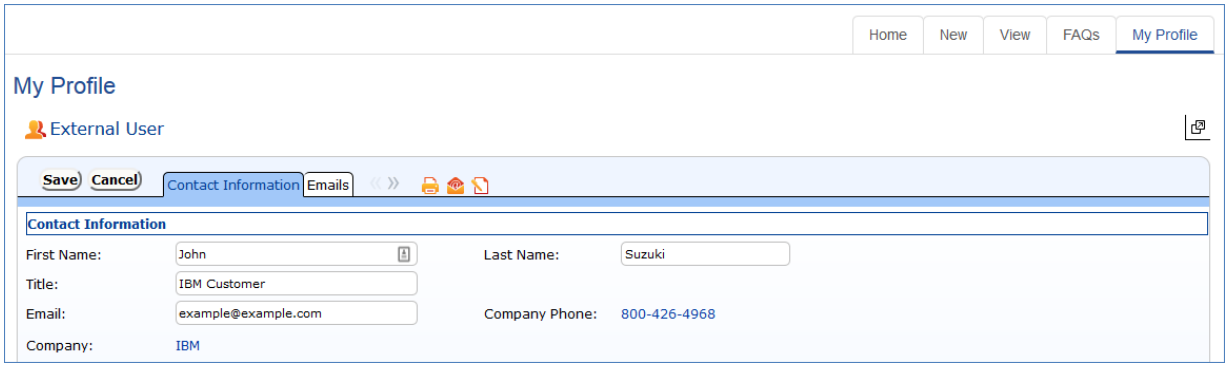


Default End User Interface (continued)

- Users in the Customer group will see links to the Support Cases they've submitted. These links exist on the Home tab and the drop-down View menu.

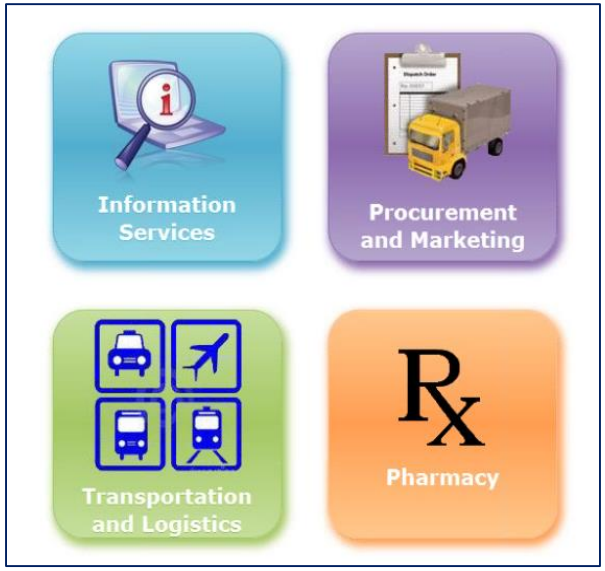


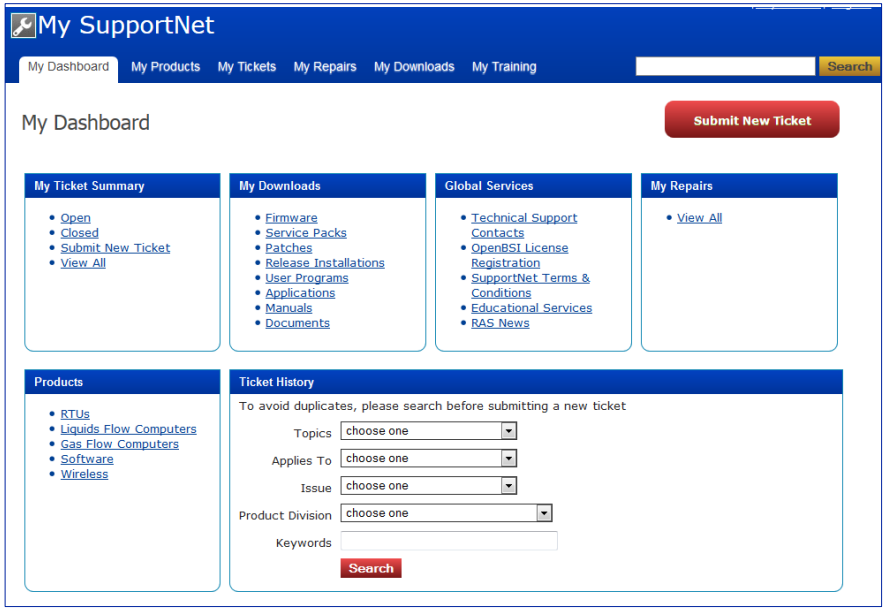
- Users can edit their contact information by clicking on My Profile. Users in the guest group will see a New User tab instead, where they can self register.



How it Works

- Agiloft's EUI is managed by a set of HTML files, stored as records in the EUI Templates table.
- Using HTML, you can redesign the interface in any number of ways.
- Some of our customers have completely overhauled their end user interface to match their organization's look and feel or to suit their business needs:



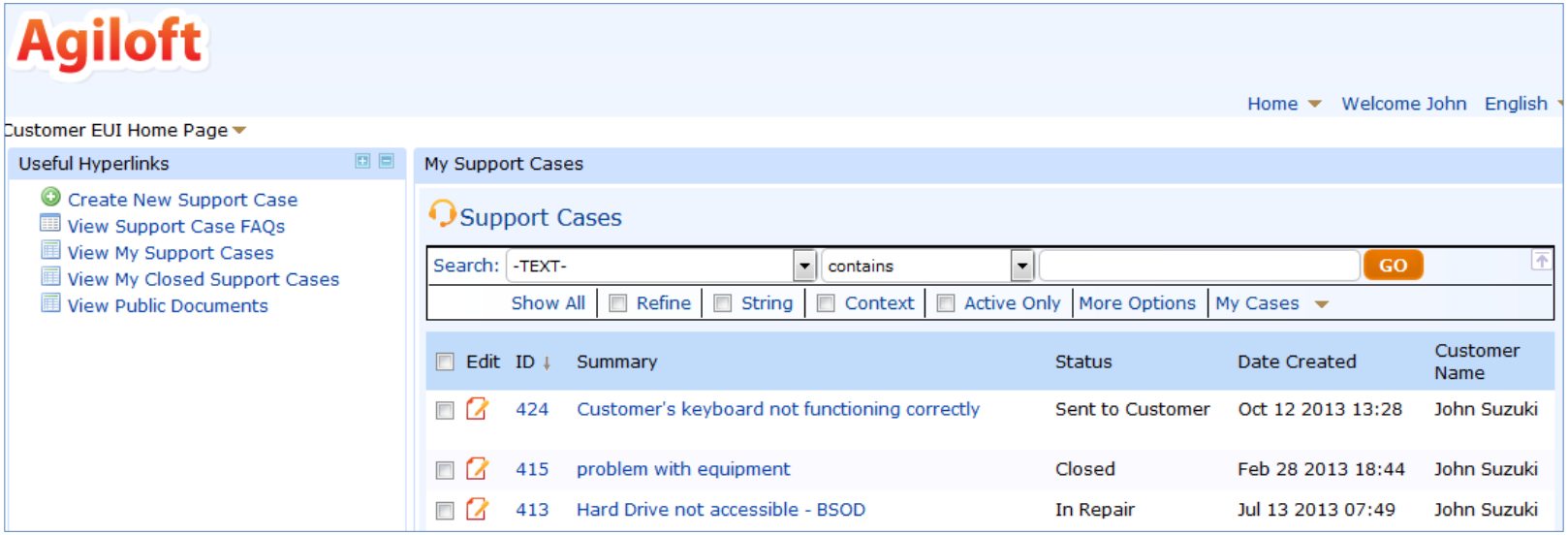


Two Interfaces: EUI vs. Legacy EUI

- Note that you have a choice of end user interfaces. In addition to the default EUI, we offer a second simple end user interface, the **Legacy EUI**, which may be appropriate for customers who do not need the flexibility provided by access to the HTML.
- Both the EUI and the Legacy EUI can be configured to show different tabs, tables, and forms depending on the user's group membership and group permissions.
- The primary EUI is customized by editing HTML code in the EUI Templates table, and it takes a bit more work to set up initially than the Legacy EUI, depending on how varied your end users are and whether you want to add new custom tables and menu options.
- Access the Legacy EUI from your custom login page by logging in as an **end user** and selecting the Staff interface radio button.
- The next slide shows the default landing page of the Legacy EUI.

Legacy EUI

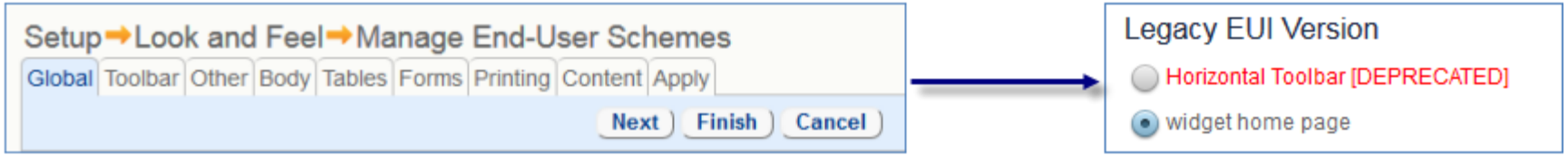
- Legacy EUI landing page, Widget-Based Home Page:



Legacy EUI Versions

- The **Legacy EUI** is controlled through the Look & Feel wizard and group permissions.
- To change the Legacy EUI Version, navigate to **Setup > Look & Feel > Manage End-User Schemes** and edit the applied scheme (Light Blue in most cases).
  - On the Global tab of the Look & Feel wizard, select Widget-Based Home Page.
  - The Widget-Based Home Page version works just like a Staff Home Page, covered in Unit 16, with a few minor differences in the features available to end users. With this style of interface, end users access all functions in the system through links and widgets.

⚠ Note that the Horizontal Toolbar option has been deprecated, as pictured below.



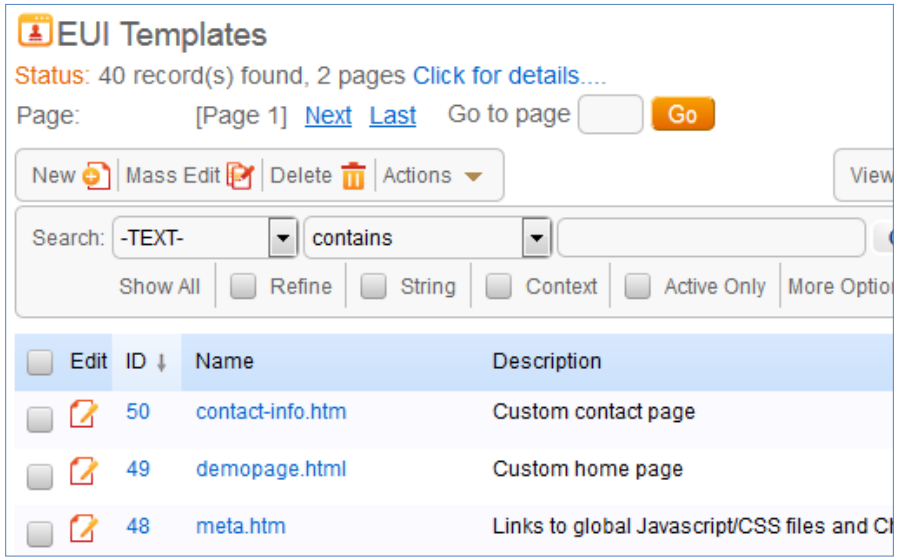
Final Legacy Interface Notes

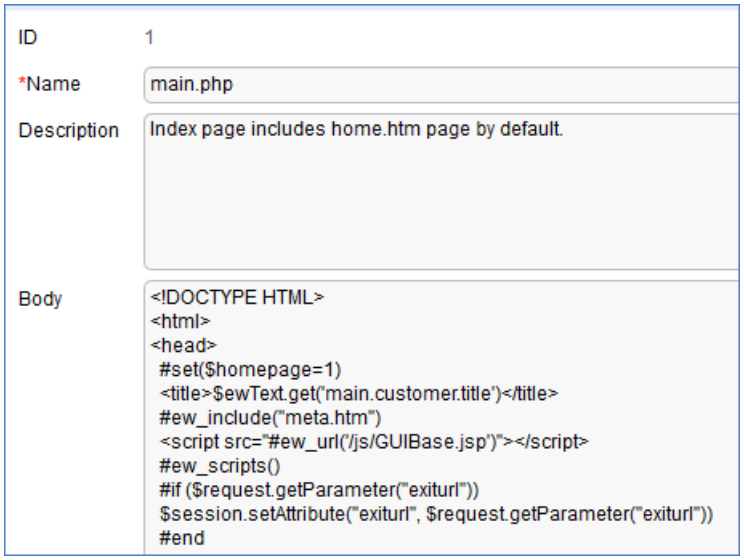
⚠ Please note that the Legacy EUI is no longer updated and will not be enhanced, but we do continue to add new functionality to the main EUI.

- The remainder of this unit pertains only to the HTML-based End User Interface, managed in the EUI Templates table. For further documentation regarding the Legacy EUI, please consult the Administrator Reference Manual.

EUI Templates Table

- Let's return to the EUI Templates table, where admins can edit the pages of the EUI and where most customization is done.
- Each page of the EUI has a corresponding record in the EUI Templates table.
- The Name field contains the record or template name and has an appropriate file extension such as .htm, .html, .css, .php or .inc.
- The Body field holds the HTML code itself.

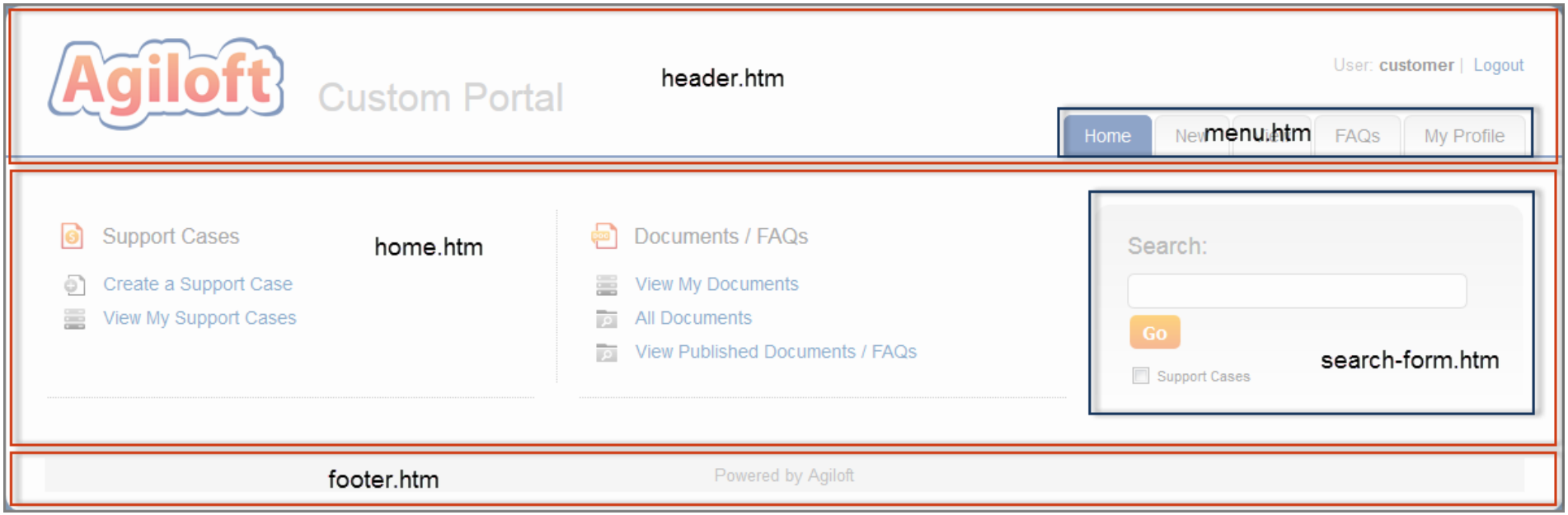




EUI Templates Table (continued)

- Some EUI Templates don't stand alone as separate pages, but are used by other templates or included in other pages.
- For instance, the templates header.htm, footer.htm and menu.htm are included in every page of the EUI.
- Other examples of these "helper" files include:
  - iresize.js, which contains a piece of JavaScript;
  - search-form.htm, which contains the search block found on the home page;
  - meta.htm, which holds meta information included in every page.
- On the next slide you will find a *wireframe*, a simple webpage schematic, to help you visualize the structure and component files of the EUI landing page, main.php.

Wireframe of main.php



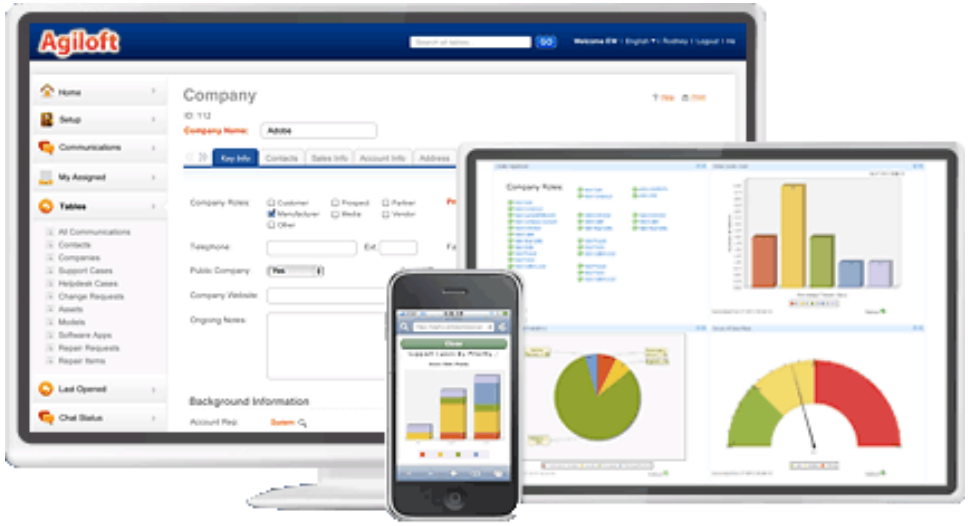
Wireframe Explained

- What does the wireframe tell us about the landing page of the EUI?
- The file main.php does not directly contain the elements that are displayed to the user on the page.
- Instead, main.php uses the macro **#ew_include()** to pull in the contents of three other HTML files (we'll learn more about macros in a few slides):

  header.htm
  home.htm
  footer.htm

- Header.htm in turn includes the file menu.htm, while home.htm pulls in search-form.htm. Header.htm also contains the reference to the logo, plus user information and a logout link in the upper-right corner.

Adding Images, Tables, and Functions



Macros and Velocity

- **Macros** are short, reusable pieces of code that represent a longer set of instructions for the software or browser to perform. Macros allow developers to save time and simplify their code.
- In the Agiloft context, macros are shorthand instructions for linking in pages, requesting data from Agiloft, or creating table views in the EUI.
- Agiloft uses the Velocity Template Language, a Java-based template engine, to expand macros (a.k.a. *Velocimacros*). For more information please visit the Velocity user guide at http://velocity.apache.org/engine/devel/user-guide.html
- Macros may require one or more additional parameters, for example:
  - #ew_forward($templateName) is a macro that creates a link to another template page in the EUI and requires the $templateName parameter. In practice, the macro might look like: #ew_forward("main.php") or #ew_forward("new_user.html")

Note that the remainder of the unit assumes prior HTML knowledge. If you need a refresher or don't have much web design experience, please refer to the HTML introduction unit.

Macros and Velocity (continued)

- More generally, macros come in the form:
  #macro_name($param1, $param2, $param3, …)
- When used in practice, each parameter is enclosed in double-quotation marks. Let's look at the **#ew_image** macro:
  - `#ew_image($imageName)` `#ew_image("/gif/photo.jpg")`
  - The **$imageName** parameter is the relative file path to the image from within the KB. Note that it's the same information provided by the **src** attribute in a linked image tag <img />.
- We'll use the **#ew_image** macro in just a few slides.

Adding Images and Customizing the Logo

- One of the first changes many customers want to make is to add their own logo to the EUI. Based on the wireframe, we now know that the logo is linked from within the header.htm file, so that is the EUI Template we will be editing.
- Before we get to the HTML, we will first need to upload our new logo into Agiloft. Once the image is in the system's /gif directory it will be available for use by the **#ew_image** macro.
- Over the next few slides we will learn how to change the logo in the EUI in the following steps:
  - Upload an image through the Look & Feel wizard.
  - Edit header.htm in the EUI Templates table, using #ew_image to link in the new logo.

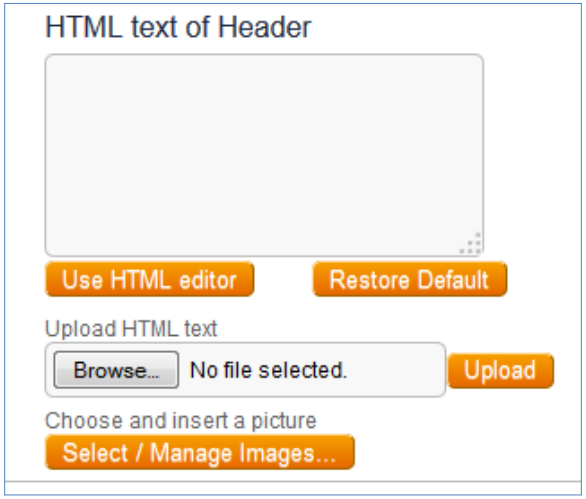
You can download the generic sample-logo.png file made available with this unit, or use any other image file you have on hand.

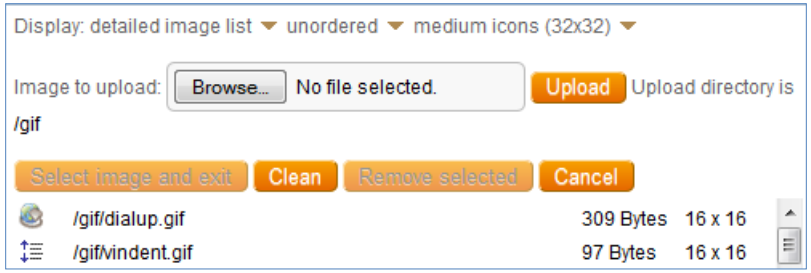
Uploading Images

- To begin, navigate to **Setup > Look & Feel > Manage End-User Schemes > [*Select scheme*]**.

We will not actually be editing this scheme, but rather using the access to the image directory. You can edit the currently applied scheme or any unused L&F scheme.

- On the Content tab, click on any of the Select / Manage Images… buttons.
- In the pop-up wizard, browse for and Upload the file sample-logo.png (or other image file).





- Click Cancel to exit the image manager. Your logo will remain in the /gif directory and can now be referenced by the #ew_image macro.
- Click Cancel again to exit the Look & Feel wizard.

Changing the Logo

- Next, navigate to the EUI Templates table and open the record header.htm.
- In the Body field, locate the line that reads:

```
<a href='#ew_forward("main.php")'><img src="#ew_image('/gif/agiloft-logo.png')"
width="185" height="75" alt="Agiloft" style="vertical-align: middle" /></a>
```

- To link in a new logo, simply replace the reference to agiloft-logo.png in the #ew_image macro with the new file name, sample-logo.png.
  - Change the **alt** attribute value from "Agiloft" to "Company logo".
  - Change the width property to "225" and the height to "100", the sample logo's dimensions. Or, leave out the height and width parameters to display any image at its full size.

```
<a href='#ew_forward("main.php")'><img src="#ew_image('/gif/sample-logo.png')" width="225"
height="100" alt="Company logo" style="vertical-align: middle" /></a>
```

- Click Save when you are finished. Log into your EUI to view the changes.

What does **#ew_image** really do? The Velocity engine expands the relative filepath, /gif/sample-logo.png, into a fully formed URL. The expanded URL becomes the value of the src attribute and links in the image.

Adding Table and Record Functions

- Adding more complex functions to the EUI is done in much the same way as adding an image: using specialized macros to add tables, records, and forms. The main difference is that **#ew_image** requires a single parameter, **$imageName**, while the more complex macros require several parameters.
- In the default EUI setup, each of the basic page types corresponds to an Agiloft macro that links in the table, record, or form and can apply search filters or set additional parameters.
  - **Table views** use the **#ew_table** macro.
    Example: My Support Cases page or All Documents page.
  - **Record editing** pages use the **#ew_edit_record** macro.
    Example: My Profile page.
  - **Submission forms** use the **#ew_create_record** macro.
    Example: Create a Support Case.
  - FAQ pages use the **#ew_faq** macro.
- We will start with a closer look at a few of these macros, and then learn how to add links between pages and further customize the EUI.

Adding Tables

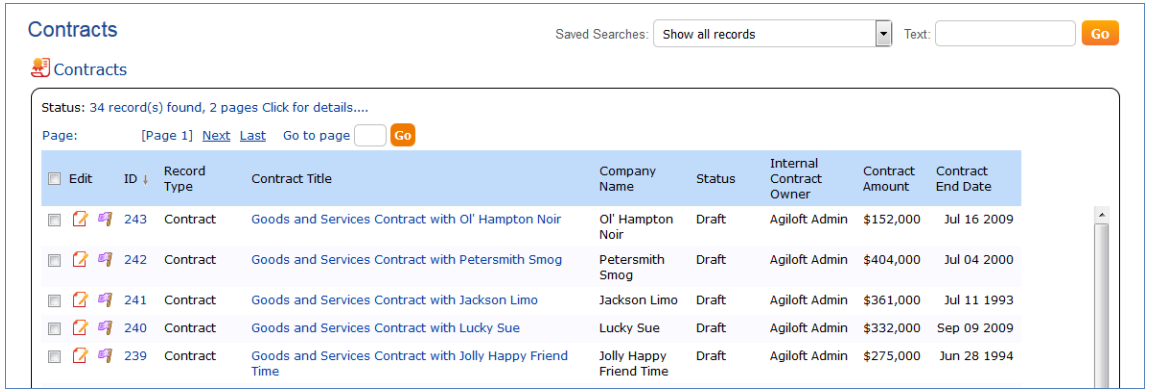
- Let's begin with the **#ew_table** macro, used in the EUI for My Support Cases, My Documents, Documents, and other pages that display a table view.
- In brief, **#ew_table** creates an **<iframe>** and displays an Agiloft table. An iframe, or *inline frame*, allows you to embed one HTML page into another.
- **#ew_table** requires the following parameters:
  - **$subtypeName** is the **logical name** of the table to display.
  - **$viewName** specifies which **view** to use to display the table. If left blank, the user's default view is applied.
  - **$searchName** is the name of a **saved search** to apply to the table. This can be left blank ("") to display all records the user has access to.
  - **$queryString** can apply additional search filters to the resulting table.
  - **$params** allows additional visualization parameters to fine-tune the display of the table. The default options are listed in the EUI Tutorial at https://wiki.agiloft.com/display/HELP/EUI+Portal+Tutorial.
  - **$frameName** is the required name of the iframe which will display the table. We use "my_table" as a default.
  - **$frameStyle** can be used to specify the CSS class name of the frame and apply additional styles, or left blank.

Edit the All Contracts Page

- Before we continue, log in to your training KB EUI as a test user with the login/password of contractuser/contractuser. This is a test user in the Contract Creator group.

Alternately, you can create a new user in the People table and use the lookup icon next to Groups to select Contract Creator.

- Once logged in, select **View > All Contracts** to see the default starting point:



- Next we will make a few edits to the macro in all_contracts.html, without changing any of the surrounding HTML.

Edit the All Contracts Page (continued)

- As the admin user, open the EUI Template all_contracts.html and locate the **#ew_table** macro in the Body field:

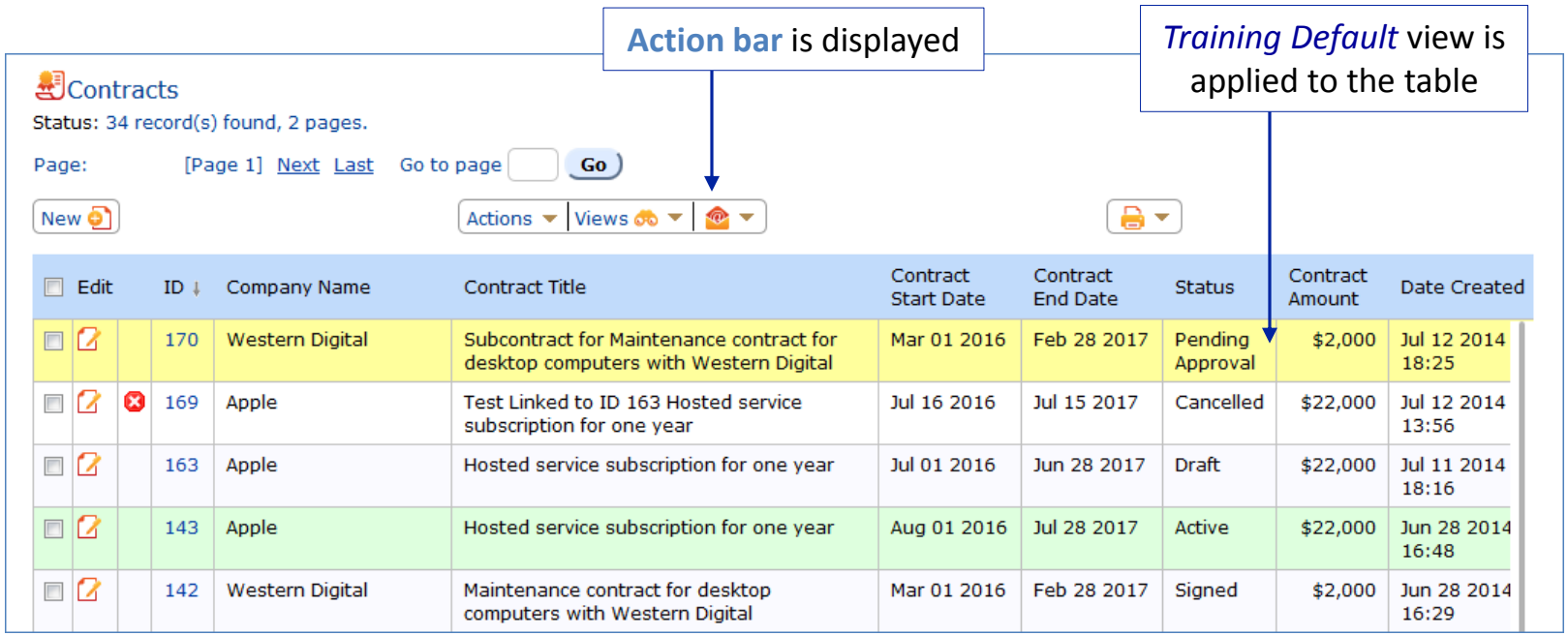
```
#ew_table("contract" "Default View" "" ""
"showToolBar=false&showNavigation=true&showfastsearch=false" "my_table" "")
```

- Change the view name (second parameter) from "Default View" to "Training Default", the view we created in Unit 2.
- The third parameter ($defaultsearch) is blank by default. You may apply any saved search from the Contracts table if you wish.
- The string showToolBar=false… are visualization parameters ($params). Let's show users the action bar by changing "showToolbar=false" to "showToolbar=true".
- The revised #ew_table macro is as follows:

```
#ew_table("contract" "Training Default" ""
"showToolbar=true&showNavigation=true&showfastsearch=false" "my_table" "")
```

Edit the All Contracts Page (continued)

- Save the EUI Template, then log back into the EUI to see the changes we made. (You can also refresh the page by selecting **View > All Contracts**.)



Editing Records

- Next let's look at the **#ew_edit_record** macro, used in the template my_profile.html to create the My Profile page in the EUI.
- **#ew_edit_record** requires only four parameters:
  - **$subtypeName** is the logical name of the table the record belongs to, just like the **#ew_table** macro.
  - **$recordId** is the ID of the target record to edit.
  - **$returnURL** specifies the URL to go to if the user presses Save or Cancel to exit the record. For EUI templates we only need the relative path, such as "/eui2template/main.php", to redirect users back to the landing page.
  - **$returnFrame** specifies the target frame for the associated $returnURL. Use "parent" as the default or when redirecting to a page with no frames.

Editing Records: My Profile Example

- How does the macro look in practice? In the Body of my_profile.html, we see:

```
'#ew_edit_record("contacts" "#ew_userid()" "/eui2template/main.php"
"parent")'
```

  - To display a user record, we set the **$subtypeName** parameter to "contacts" to reference the People table.
  - To set the **$recordId** parameter, we need to find the current user's ID number. To accomplish this we use another macro, **#ew_userid()**, which retrieves the current user's ID and passes the value to #ew_edit_record.

    #ew_userid() requires no parameters.
  - Users are redirected to the landing page by setting the **$returnURL** parameter to "/eui2template/main.php".
  - Again, the default "parent" is used for the **$returnFrame**.

Creating Records

- Our macro for creating records and submission forms is **#ew_create_record**, used in the Create a Support Case (new_supportcase.html), Create a Contract (new_contract.html), and other record submission pages.
- The **#ew_create_record** parameters are identical to those in **#ew_edit_record**, without the need for the **$recordId** parameter.
  - **$subtypeName**
  - **$returnURL**
  - **$returnFrame**
- We have now seen how **#ew_table**, **#ew_edit_record**, and **#ew_create_record** are used to provide Agiloft functions within the HTML environment of the EUI templates.
  - We've also covered the macros **#ew_image** and **#ew_userid**.
- In addition to these specialized macros, we use other features of the Velocity Template Language to build responsive webpages. Over the next slides we will discuss the use of *conditionals* and other *methods*.

Conditionals

- The **Velocity Template Language (VTL)** has a prebuilt **#if** *directive*. Used within the body of an EUI Template, **#if** helps to apply conditional visibility to various HTML elements.
- We won't cover the details of VTL syntax here, but rather jump right in. Additional documentation can be found in the Velocity user guide at http://velocity.apache.org/engine/devel/user-guide.html.
- Let's consider a simple example:

```
#if (condition A is true)
<p>Welcome!</p>
#end
```

- This will display the text 'Welcome!' only if the condition is true; otherwise nothing happens.

Conditionals (cont.)

- To set up more complex conditions, you can use the full range of If/ElseIf/Else logic.
- You can also use relational and logical operators within an **#if** condition:

Operator	Definition
||	OR
&&	AND
!	NOT
<	Less than
>	Greater than
==	Equal/equivalent

- For example:
  - #if ([conditionA] && [conditionB])
    This will only evaluate if both conditions A and B are true.
  - #if ([conditionA] || [conditionB])
    This will evaluate if condition A or condition B is true (or both).
- So how do we write a useful condition in the template?

User Related Tools

- A Velocity *method* is defined in Java and can be used to check conditions or arrive at decisions.
- Our next practice exercise makes use of a specialized method, **$ewUser.isInGroup**, one of a family of user-related tools in Agiloft.
- Using the format **$ewUser.isInGroup("group name")**, this method checks the current user's group membership and returns 'true' if the specified group name is found.
- Similarly, we can use **$ewUser.isInTeam("team name")** to verify a user's team membership.
  - Remember you can use operators to combine conditions. For example:

```
#if ($ewUser.isInGroup("Internal Customer") || $ewUser.isInGroup("Customer"))
```

- For example, the code below will display a welcome message only to users in the Guest group:

```
#if ($ewUser.isInGroup("Guest"))
<h2>Welcome, new user!</h2>
#end
```

Conditional Branding

- Now suppose you would like to display a different logo to users in a particular group or team. With a few simple statements in the EUI Template header.htm, we can achieve this goal.
- For our practice exercise we are going to display the Agiloft logo to members of the Contract Creator group, and our new sample logo to all other end users.
- In the header.htm record, locate the line we edited on a previous slide:

```
<img src="#ew_image('/gif/sample-logo.png')" width="225" height="100" alt="Company
logo" style="vertical-align: middle" />
```

- Replace this with the following lines of code and Save the record:

```
#if ($ewUser.isInGroup("Contract Creator"))
   <img src="#ew_image('/gif/agiloft-logo.png')" width="185" height="75" alt="Agiloft"
style="vertical-align: middle" />
#else
   <img src="#ew_image('/gif/sample-logo.png')" width="225" height="100" alt="Company
logo" style="vertical-align: middle" />
#end
```

Conditional Branding (continued)

- In simple language, we can translate the code as follows:

  If the user is in the Contract Creator group
  show the image agiloft-logo.png
  Otherwise (for all other users)
  show the image sample-logo.png
  #end (to close off the statement).

- Log in with different test users to confirm that only contract creators can see the initial Agiloft logo.

- We will see the method $ewUser.isInGroup again when we work with menu.htm to create a new menu tab.

Creating a New Page

- Next we will use the **#ew_create_record** macro to create a record submission form for the Tasks table.
- The basic steps for adding a new page to the EUI are:
  - Create the HTML template.
  - Add a link from the menu and/or home page.
  - Set user permissions and create an **end user layout**.
- As we complete the practice exercise we will cover additional topics, including:
  - Creating links with **#ew_forward**.
  - Making changes to the menuChange JavaScript function in menu.htm.
  - Using **$ewText.get** and translation.properties.

Creating a New Page in the EUI

- We are going to add a page to the EUI that gives end users the ability to submit a task—in other words, to create a record in the Tasks table.
- To get started, navigate to the EUI Templates table.
- Copy the template new_user.html to use as a starting point. When prompted, enter the name “new_task.html” for this template record. Click Finish.
- Open new_task.html for edit and change the Description to “Page for creating new tasks in the EUI.”
- Now take a look at the <iframe> tag, which contains the **#ew_create_record** macro:

```
<iframe src='#ew_create_record("contacts.employees" "/eui2template/main.php"
"parent")' name="content_frame" id="frameres" width="100%"
frameborder="0"></iframe>
```

- Replace contacts.employees (the first parameter) with "task" (the logical table name of the Tasks table). We can leave the second and third parameters as-is, which will redirect users to the landing page after submitting the form or pressing Cancel.
- Click Save. We will return to edit the new_task.html template in a few slides.

Working with menu.htm

- The next step in making our new page accessible is to add a link from the menu by making a few changes to menu.htm.
- Each menu tab is a list item <li> containing a link <a> to a template page.
- Menu.htm makes heavy use of *conditionals* (if-else statements) and the *method* **$ewUser.isInGroup** to ensure that different types of end users see only the tabs relevant to them.
- In our initial setup, the EUI is configured for users in several different groups. The table here outlines the available functions and navigational tabs each user group sees when accessing the EUI.
- Any groups not listed (such as our test user in the Customer group) will see the following tabs:
  Home; New (Support Case); View (Support Cases);
  FAQ (Support Cases); and My Profile.

Group	Default EUI Tabs
Guest	• Home • New (Support Case) • View • My Profile

Internal Customer	• Home
	• New (Helpdesk Case)
	• View
	  ▪ My Helpdesk Cases
	  ▪ My Documents
	  ▪ All Documents
	• FAQs
	  ▪ Helpdesk Cases
	  ▪ Documents
	• My Profile

Document Creator	• Home • New (Document) • View (My Documents) • FAQs (Documents) • My Profile

Contract Creator	• Home
	• New (Contract)
	• View
	  ▪ My Contracts
	  ▪ All Contracts

Repair Customer	• Home
	• New (Support Case)
	• View
	  ▪ My Support Cases
	  ▪ My Documents
	  ▪ All Documents
	• FAQs (Support Cases)

Working with menu.htm (continued)

- Let's look at the first <li> item in menu.htm:

```
<li id="main"><a href='#ew_forward("main.php")'>$ewText.get("menu.home")</a></li>
```

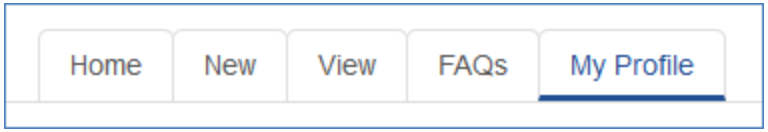
- There are a few things to note about the code. First, the link's href attribute is formed with the **#ew_forward** macro.
  - As we have seen, this macro forwards the user to the page specified by the **$templateName** parameter.
  - Using **#ew_forward** keeps the user logged in to Agiloft while navigating between pages.

```
<!--Menu-->
    <div class="menu-wrapper">
     <ul class="menu">
      <li id="main"><a href='#ew_forward("main.php")'>$ewText.get("menu.home")</a></li>
#set($tableeditcount=0)
#if ($ewText.get("show.menu.new")=="yes")
      <li id="new" class="submenu"><a>$ewText.get("menu.new")</a>
       <ul>
#if ($ewText.get("show.helpdesk.cases")=="yes" && $ewPermission.table("helpdesk_case","create"))
        #set($tableeditcount=$tableeditcount+1)
        <li><a href='#ew_forward("new_helpdeskcase.html")'>$ewText.get("menu.new.hc")</a></li>
#end
```

- Second, the link text is referenced by the method **$ewText.get()**. We will return to **$ewText.get** in more detail a little later.
- Last, each list item is assigned an **id**. For the Home tab link above, the **id** is "main". Each menu tab is assigned an id that refers to some part of the URL (i.e., main.php). (The id refers to the URL for convenience' sake—it is perfectly fine to name the id 'bananas' as long as you are consistent.)

Highlighting the Active Menu Tab

- The menu tab's **id** is used in part by the JavaScript function, menuChange(). This simple script allows the browser to recognize the current page and highlight the corresponding menu tab by changing the underline color:



- To follow along, open menu.htm and scroll to the bottom of the Body field. Locate the <script> section and the block of text beginning function menuChange().

- This piece of JavaScript looks at the browser's current URL. If it finds the string 'main.php' in the URL, it assigns the 'active' class (a CSS reference) to the element on the page with an id of 'main'.

```
function menuChange() {
  if (document.getElementById && document.createTextNode) {
    if (homepage || location.href.match('main.php')) {
      document.getElementById('main').className='active';
    } else  if (location.href.match('new_user')) {
      document.getElementById('new_user').className='active';
    } else  if (location.href.match('new_')) {
      document.getElementById('new').className='active';
    } else if (location.href.match('-faq')) {
      document.getElementById('faq').className='active';
```

- If 'main.php' is not in the URL, it goes through the remaining conditions in the script. Next, it looks for the URL string 'new_user' to find a match, followed by 'new_' and so on.

Linking from the Menu

- The stylesheet (style.css) holds a rule that sets the background color of an ‘active’ menu tab. Later in this unit we will adjust this color property in the stylesheet.
- Now let’s create a link to new_task.html. From the EUI Templates table, open menu.htm for edit (if not already open).
- Copy lines 8-10, the helpdesk cases link, onto their own lines immediately below the originals.

```
#if ($ewText.get("show.service.requests")=="yes" &&
$ewPermission.table("helpdesk_case","create"))
<li><a
href='#ew_forward("new_servicerequest.html")'>$ewText.get("menu.new.sr")</a></li>
#end
```

Linking from the Menu (continued)

- Change $ewText.get("show.service.requests") to $ewText.get("show.tasks")
- Change $ewPermission.table("helpdesk_case", "create") to ("task", "create").
- In the #ew_forward macro, change "new_servicerequest.html" to "new_task.html".
- Change the link text, $ewText.get("menu.new.sr") to Task.

  ⚠ We will return to **$ewText.get()** shortly!

- The revised lines of HTML are:

```
#if ($ewText.get("show.tasks")=="yes" && $ewPermission.table("task","create"))
<li><a href='#ew_forward("new_task.html")'>Task</a></li>
#end
```

- Click Save.

Task Table Permissions

- The next step in creating our task submission page is to create an **end user layout** and give appropriate users access permissions to the Tasks table.
- Navigate **to Setup Tasks > Layout**. Check if an end user layout exists; if not, copy the staff layout and make additional changes if desired. Click Finish to save the changes.
- Next, navigate to **Setup > Access > Manage Groups**, then edit the Customer group.
- On the Table tab, find and edit the Task table.
- On the Record Permissions tab, allow the group to Create Tasks, and View their own Tasks.
- On the Field Permissions tab, make your selections match the screenshots on the following slide. Think about what each selection means for the user.
- When you are done, click Finish in the Table Permissions wizard and then Finish again in the Manage Groups wizard.
- View the completed task submission page in the EUI.

Task Table Permissions

Field	View own	Create	Edit own	View other	Edit other
Add Time	☐	☐	☐	☐	
Alt Status	☐	☐	☐	☐	
Assigned Person	☑	☑	☐	☐	
Assignee ID	☐	☑	☐	☐	
Assignee Manager	☑	☑	☐	☐	
Assigned Team	☑	☑	☐	☐	
Assigned Team Leader	☐	☑	☐	☐	
Attached Files	☑	☑	☐	☐	
Billable	☑	☑	☐	☐	
Billable Amount	☑	☑	☐	☐	
Billable Hours	☑	☑	☐	☐	
Billable Rate	☑	☑	☐	☐	
CC List	☑	☑	☐	☐	
Closing Comment	☐	☐	☐	☐	
Communications	☐	☐	☐	☐	
Created By	☐	☐	☐	☐	
Creator Login	☐	☐	☐	☐	
Creator Team	☐	☐	☐	☐	
Date Completed	☐	☐	☐	☐	
Date Created	☐	☐	☐	☐	
Date Due	☑	☑	☐	☐	
Date Updated	☐	☐	☐	☐	
Deletable	☐	☐	☐	☐	
Demo Data	☐	☐	☐	☐	

History	☐	☐	☐	☐
ID	☑	☑	☐	☐
Mark as Completed	☐	☐	☐	☐
Project Category	☑	☑	☐	☐
Project ID	☑	☑	☐	☐
Project Manager	☑	☑	☐	☐
Project Name	☑	☑	☐	☐
Project Status	☑	☑	☐	☐
Related Assets	☑	☑	☐	☐
Related To	☑	☑	☐	☐
Send Notes to Creator	☐	☐	☐	☐
Status	☑	☑	☐	☐
Task Description	☑	☑	☐	☐
Task Summary	☑	☑	☐	☐
Task Type	☑	☑	☐	☐
Time Description	☐	☐	☐	☐
Time Entries	☐	☐	☐	☐
Time Spent	☐	☐	☐	☐
Total Time Spent	☐	☐	☐	☐
Type	☐	☐	☐	☐
Updated By	☐	☐	☐	☐
Updater Login	☐	☐	☐	☐
Updater Team	☐	☐	☐	☐
Working Hours to Complete	☐	☐	☐	☐
Working Notes	☐	☐	☐	☐

Permissions and the End User Interface

- As a review of previous units, let's discuss some important ways that group permissions affect the final product—that is, exactly what end users see on an EUI page.
- On the next slide we will compare the form a user sees when submitting a Support Case to the form they see when viewing an existing support case (**View > My Support Cases > [Select any case to view]**).
- Fewer fields are displayed via **New > Support Case**. Note that none of the customer's contact information appears, nor does the Status field, Date Created, Date Updated, or the Emails tab.
- This is because the Customer group has **create permissions** for limited fields on the Support Cases table. The group has more extensive **view permissions**, which apply to records viewed after creation, such as those accessed from **View > My Support Cases**.
- View/edit/create permissions, together with the end user layout settings (**Setup [Table] > Layout**), define what the end user actually sees.

Permissions and the EUI (continued)

Create Permissions

Submit a Support Case

Support Case

Save Cancel Details

ID: 429 Priority: Low

Problem Description

Type of Issue: Question

*Summary:

Problem Description:

Inbound Attachments: Attach/Manage

Drag&Drop files

Save Cancel

View Permissions

Support Case

Edit Close Details Emails

ID: 424 Priority: Low Status: Sent to Customer

Contact Information

*Customer Name: John Suzuki

Customer Phone: 333-555-1212 Ext.122 Cell Phone:

*Customer Email: example@example.com

Problem Description

Type of Issue: Equipment Problem

*Summary: Customer's keyboard not functioning correctly

Problem Description: Customer's keyboard has function keys that do not work.

Additional Notes: [Anonymous User Jun 10 2013 06:52]
Here is some additional information.
[John Suzuki Jun 10 2013 06:58]
Here is my reply.

Inbound Attachments: leftpanetables.gif Downloadable Files:

Solution:

Date Created: Sep 12 2015 13:28 Date Updated:

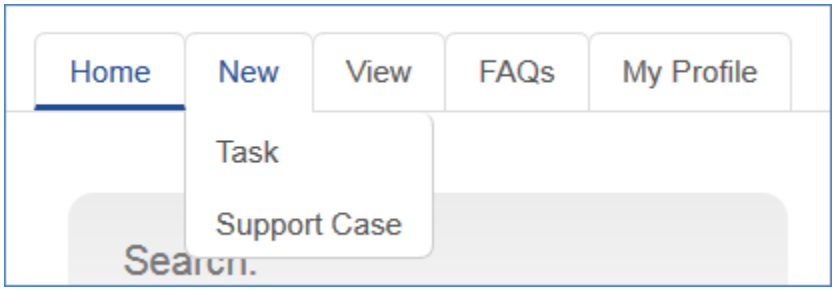
Edit Close

The take-away is that if you don't immediately see what you expect, check your permission settings!

Agiloft

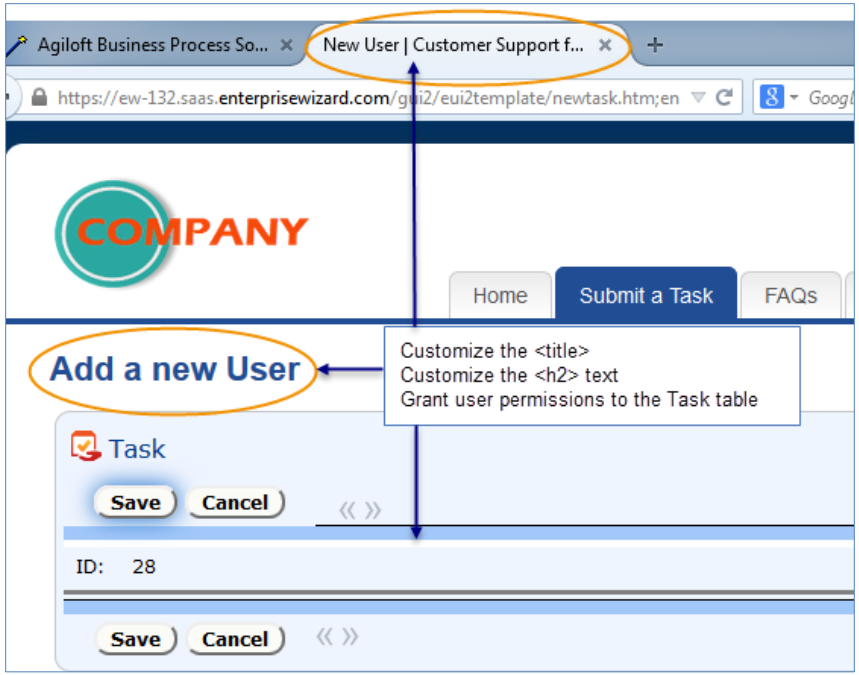
Edit $ewText.get

- In the menu.htm file, we've used conditional logic to determine whether or not to show the Tasks table. The code checks to see if show.tasks=yes in the translation.properties file by using $ewText.get to retrieve that piece of information. Therefore, for our new Task links to appear, we need to properly set this attribute in the translation.properties file.
- To do so:
  - Edit the translation.properties file.
  - Under #These values determine whether a table will be shown at all in the end user interface…, find line that reads: show.tasks=no.
  - Change the line to read show.tasks=yes.
  - Click Save.
- Log back into the end user interface.
- The customer/customer user now sees the following menu tabs:



Custom Text in the EUI

- Now that we have the menu working properly, our next step is to customize the page title and heading text.
- Remember that we initially copied new_user.html as a starting point, so that is the text we currently see on our New Task page.
- After we customize the text, we will set group permissions and create a **layout**.
- We entered the text manually when we created the Task link, while other links used the **$ewText.get()** macro.



Localization and translation.properties

- The purpose of **$ewText.get()** is to allow for a multi-lingual environment.
- The method **$ewText.get($key)** works together with another EUI Template, translation.properties.
- Translation.properties is actually a family of templates; the default Agiloft setup includes translation_ru.properties and translation_zh.properties which hold Russian and Chinese text, respectively (the two-letter combinations are standard Java codes for *locales*).

⚠ The file translation.properties is used when no locale is set for the user's session. This is initially in English, but you could choose to use a different default locale/language.

- Each row of translation.properties consists of a **key/value** pair, in the form:

```
key1=value1
```

- One line of translation.properties reads:

```
home.1table.title=Welcome to the Agiloft Custom Portal
```

Localization and translation.properties (continued)

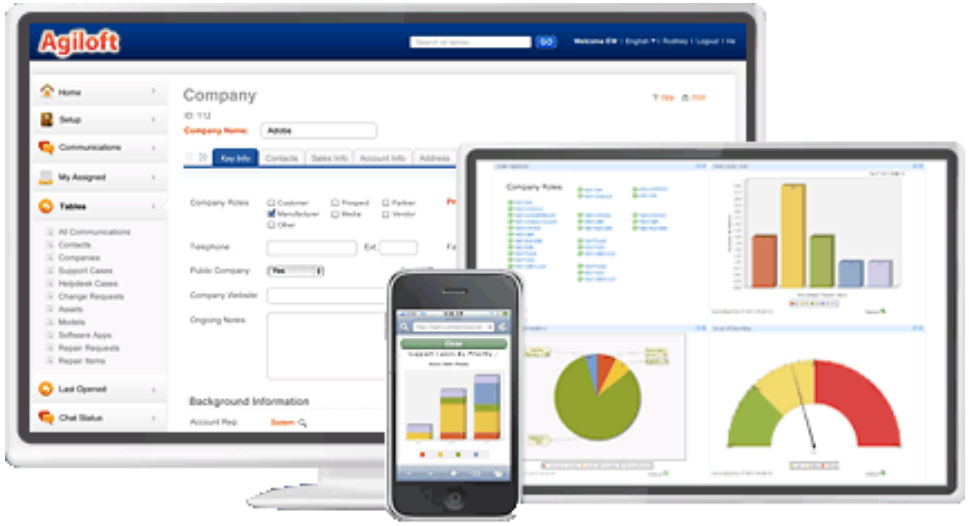
- In this case, home.1table.title is the *key*, and the *value* is the text "Welcome to the Agiloft Custom Portal."
- The **$ewText.get($key)** method checks the user's session locale and uses that to determine which translation_xx.properties file to use.
- Setting the session locale can be done in a number of ways. One possibility is to set the locale in the login link; another is to use the **#ew_languages** macro to display a list of available languages for the user.
- For our practice exercise, we will only add the new key/value pair to translation.properties. Keep in mind that in a multi-lingual environment, the key and its translated value must be added to each translation template file.

⚠ For more information on Agiloft's localization-related macros, please refer to https://wiki.agiloft.com/display/HELP/EUI+Macro+Reference.

Customizing Text

- Open translation.properties from the EUI Templates table and add:
  - The following line of text under #These variables provide the text on the home pages:
    - menu.new.task=Task
    - newtask.title=Create a Task | Agiloft End User Portal
    - newtask.header=Create a Task
- Click Save.
- Now open new_task.html and make the following changes in the Body field:
  - Within the <title> tags, replace “register.title” with “newtask.title”.
  - Within the <h2> tags, replace “Register” with “Create a Task”.
- Log back in to the EUI and note the changes.

CSS and FAQs



Agiloft

Working with Styles

- As a simple introduction to CSS we will alter a few of the style rules and change a few colors to complement the sample logo.
- Open style.css and scroll through the contents. The lengthy style instructions are broken up by their functions (i.e. Background, Header, Search form, etc.) with comment lines such as:

  /* Background */

- Use these landmarks to find the specific styles you want to adjust.
- A slash and asterisk (/*) begin a comment line in CSS, and the comment ends with an asterisk and slash (*/). You can also use the HTML comment format, which begins with <!-- and ends with -->
- The first tag below the **<body>** of each EUI page is , and this is the code responsible for the background. This means that everything under .wrapper in the style sheet applies to the background.

Changing Styles and Colors

- The default setup uses a dark blue background, overlaid by a blue-to-blue gradient background. To make things simple, let's comment out the lines relating to the background color gradient.
- Add a comment line above the line background: url(data:….:
  /****** remove gradient background*******
- End the comment after the line filter: progid:DXImageTransform…
  *****end of background gradient ******/

Add /* and */ around a section to comment it out of the CSS file.

```
  background: #003060;
/******* remove gradient background*********
  background: url(
/Pgo8c3ZnIHhtbG5zPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwL3N2ZyIgd2IkdGg9IjEv
EgMSIgcHJIc2VydmVBc3BIY3RSYXRpbz0ibm9uZSI+CiAgPGxpbmVhckdyYWRpZW5(
udFVuaXRzPSJ1c2VyU3BhY2VPbIVzZSIgeDE9IjAIiB5MT0iMCUiIHgyPSIwJSIgeTI9IjE
vcC1jb2xvcj0iIzA0NGQ5NiIgc3RvcC1vcGFjaXR5PSIxIi8+CiAgICA8c3RvcCBvZmZzZXC
Atb3BhY2I0eT0iMSIvPgogIDwvbGluZWFyR3JhZGIlbnQ+CiAgPHJlY3QgeD0iMCIgeT0
KCNncmFkLXVjZ2ctZ2VuZXJhdGVkKSIgLz4KPC9zdmc+);
  background: -moz-linear-gradient(top,  #003060 0%, #044d96 100%);
  background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#003060), co
  background: -webkit-linear-gradient(top,  #003060 0%,#044d96 100%);
  background: -o-linear-gradient(top,  #003060 0%,#044d96 100%);
  background: -ms-linear-gradient(top,  #003060 0%,#044d96 100%);
  background: linear-gradient(to bottom,  #003060 0%,#044d96 100%);
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#003060', endCol
********end of background gradient *****/
  background-repeat: no-repeat;
  background-attachment: fixed
```

Changing Styles and Colors (continued)

- In HTML and CSS, colors can be referenced by hexadecimal value ("#11A79B"), RGB value ("rgb(x,x,x)"), or by name ("navy"). The three statements below are equivalent and will set the background color to white:

```
background: rgb(255,255,255); background: white;  background: #FFFFFF;
```

  You can use a website like http://html-color-codes.info/ to convert between Hex codes and RGB values.

- Directly above the gradient section we have a *declaration* which reads: background: #003060*;*
  - Change the color to #2AA9A0 (the shade of green used in the sample logo), or any other background color you wish.
- Next let's change the color of the line under the menu tabs. In style.css, look for the text:

```
.header {
border-bottom: 1px solid #dddddd;
```

- Change the color to #F4490A. This is the orange color used in the sample logo.

Changing Styles and Colors (continued)

- Next we will change the color of the active menu tab to match the header's bottom-border. Locate the line: *.menu li.active, .menu li.active:hover {*
  - Change the background and border colors to #F4490A.

```
}
.menu li.active, .menu li.active:hover {
    text-shadow: none;
    color: #214d94;
    z-index: 2;
    border-bottom: 3px solid #214d94;
    cursor: default;
    margin-top: 0;
    padding-bottom: 0;
}
```

```
}
.menu li.active, .menu li.active:hover {
    text-shadow: none;
    color: #F4490A;
    z-index: 2;
    border-bottom: 3px solid #F4490A;
    cursor: default;
    margin-top: 0;
    padding-bottom: 0;
}
```

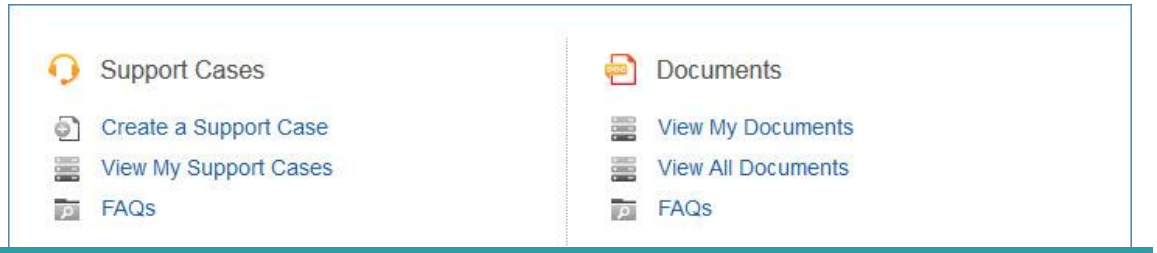
- To change the heading text color on individual pages, find h2 in style.css and replace the color with orange, #F4490A.

```
a:hover { text-decoration: underline }
h1 { font-size: 26px; line-height: 130%; color: #939393; font-weight: normal }
h2 { font-size: 22px; line-height: 130%; color: #214d94; font-weight:  normal; margin: 0 0 10px }
h3 { font-size: 18px; line-height: 130%; color: #777; font-weight: normal }
input { outline: none }
```

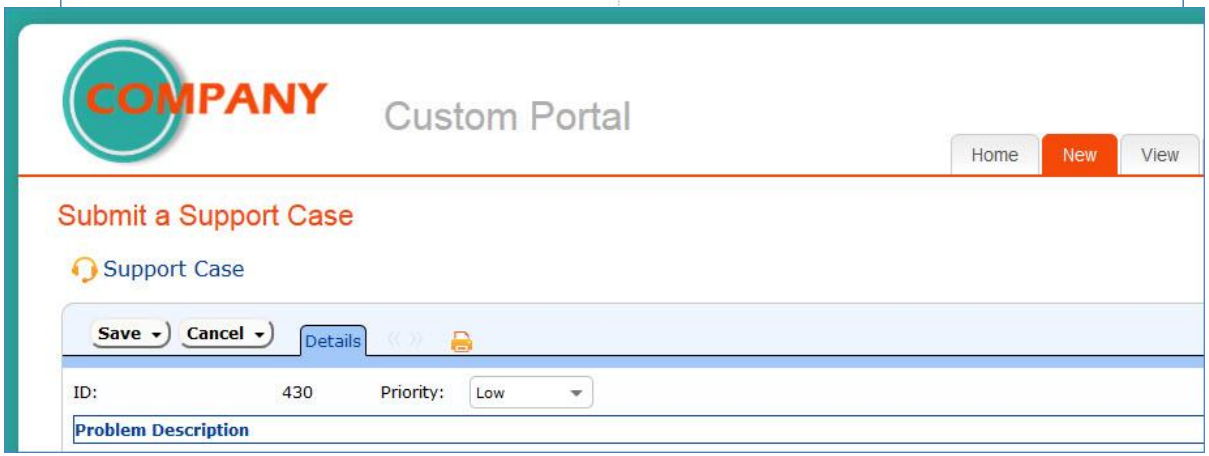
- Save the record and login to the EUI to see the new look!

Revised Styles

- Home page:



- Support Case page:



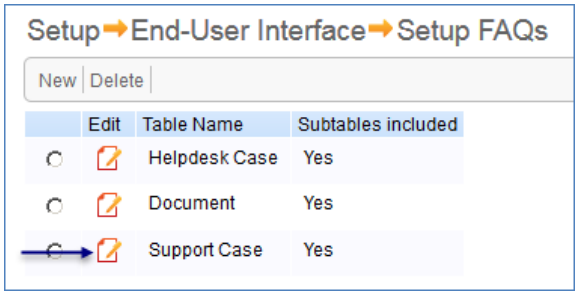
- This concludes our introduction to CSS.

FAQ Setup

- You can give your end users access to records in any table marked as FAQs in a few simple steps.
- For example, you may wish to publish solutions to common Helpdesk or Support Case issues to decrease the number of cases submitted.
- In brief, the first step is to find the records that will be published as FAQs. You can do this by creating a **saved search** with any filter you like. Then:
  - Customize the display of the FAQ search block;
  - Apply a default view;
  - Give access permissions to end users.
- The second step is to provide access to FAQs from the EUI by creating or editing a template page to display FAQs using the **#ew_faq** macro.
- On the next slides we will outline the tabs of the Setup FAQ wizard.

Tabs of the FAQ Wizard

- Navigate to **Setup > End-User Interface > Setup FAQs**. Click the Edit icon next to Support Case to access the wizard and view the FAQ setup.



- Tables: On this tab, select the table to use in the FAQ page and specify the saved search to use.
  - The Support Cases table contains a Yes/No choice field, Published, which is used by support staff to mark these special records and include them in the FAQ search.
- Display: Select up to four Search Drop-down Fields to let end users narrow their search to certain subjects before entering keywords. You may select from all choice fields in the current table.
  - Choose the default summary view to use under Show records using this view, or create a new one by clicking New.
  - You can select additional fields to display when the user opens an FAQ record by using the checkboxes under the Viewable Fields heading.

⚠ The fields selected for display will be visible to all users with FAQ access, overriding the viewer's groups permissions.

Tabs of the FAQ Wizard (continued)

- Additional options on the Display tab are used to sort FAQs and make saved searches available to end users.

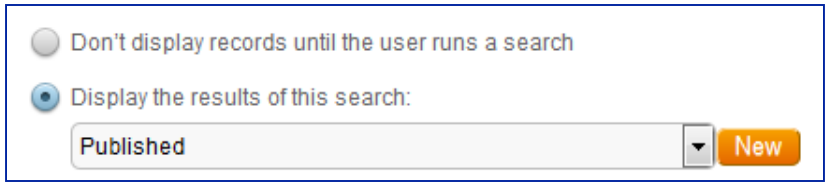
- Access: All user groups will be displayed on this tab. To enable a group's access to FAQs select All or choose to restrict FAQs further to the results of a saved search. You can create a new saved search from this tab if necessary.
  - You can also select a Rating of FAQ's scale and allow end users to submit feedback on FAQs. The rating mechanism is used for the Most Useful sorting option on the Display tab.
- Layout: Set the arrangement of the fields selected on the Display tab. The fields selected will be arranged vertically in the order selected on the Layout tab. To change the layout, drag-and-drop the field label along the row.

FAQ Setup and Permissions

- The following permissions and setup are necessary for an end user group to access FAQs through the EUI:
- Navigate to **Setup > End-User Interface > [Select FAQ to edit] > Access**.
  - Either the All or saved search radio button must be selected for the group on the Access tab of the FAQ wizard.
- Navigate to **Setup > Access > Manage Groups > [Select group to edit]**.
- On the Table tab, edit the appropriate table.
  - On the General tab, set the options Allow access to the table? and Show table on the Toolbar? to Yes.
  - On the Menu Permissions tab, allow end users to create their own saved searches by choosing the Allow for their own saved searches or Allow for all saved searches option.
  - On the Record Permissions tab, select View FAQs for [*Table Name*] the in the bottom, right corner and choose All or select a saved search.
- New FAQ configurations must be added to the EUI by creating or editing an EUI Template page. Use the **#ew_faq($subtypeName)** macro, and be sure to create a link to the FAQ page.

FAQ Setup and Permissions (continued)

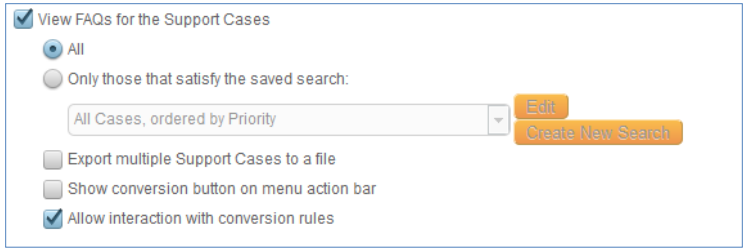
- For our practice exercise we will edit the configuration of the Support Cases FAQ and finish enabling access for end users in the Customer group.
- Navigate to **Setup > End-User Interface > Setup FAQs** and edit Support Case.
- Tables tab: select the saved search Published.
- Display: Find the option to Display the results of this search: and select Published.



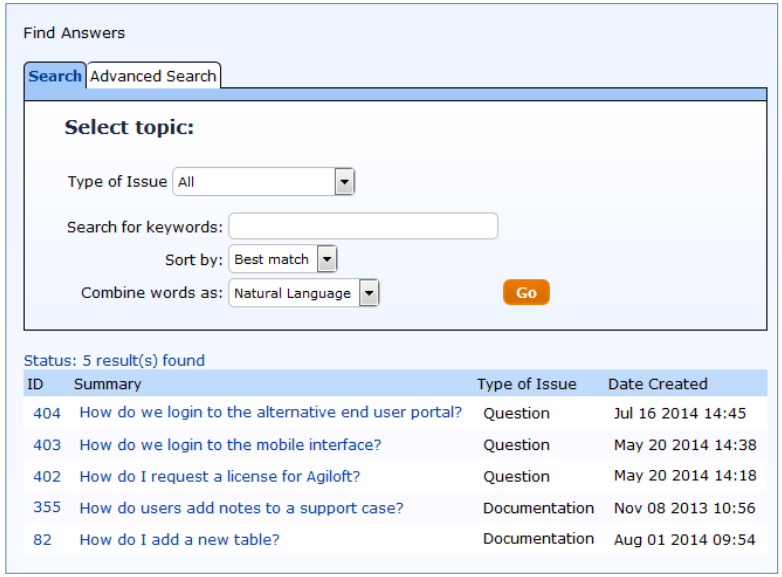
- Access: Make sure the All radio button is selected for the Customer group.
- Next, navigate to **Setup > Access > Manage Groups** and edit the Customer group.
  - On the Tables tab of the wizard select the Support Case table. In the pop-up table permission wizard, make sure the options Allow access to the table? and Show table on the Toolbar? are both set to Yes.

FAQ Setup and Permissions (continued)

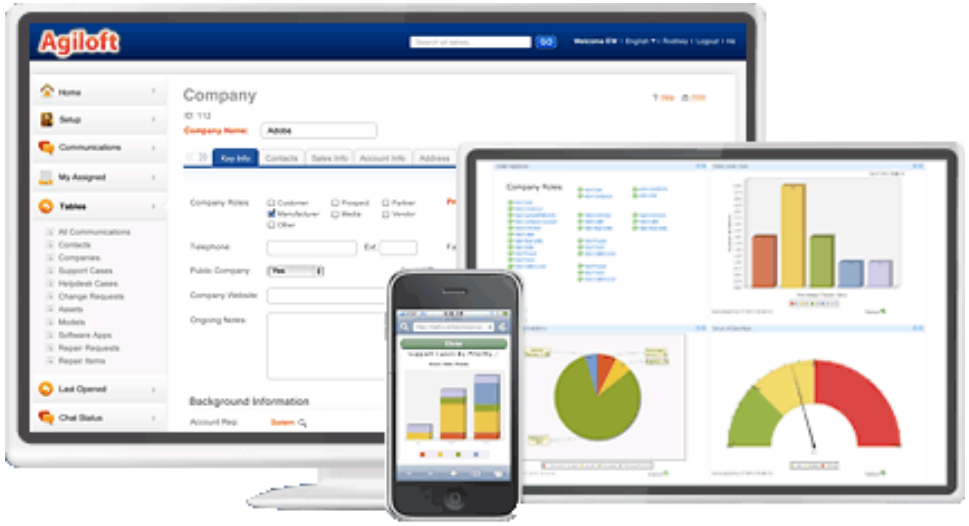
- On the Record Permissions tab, check the box to allow Customer group members to View FAQs for the Support Cases and select the All radio button.



- Click Finish to save the changes.
- Log in to the EUI as customer/customer and test the FAQ page.
- This concludes our FAQ practice exercise.



Home Page Setup: Review and Recap



Agiloft

Home.htm Setup

- The following slides are provided as a guide to the initial setup of home.htm.
- As we have learned, the landing page (main.php) includes home.htm which contains the welcome text, home page links, and search form.
- The home page is rendered with nested tables. A 1x2 table contains search-form.htm in the second cell to present the search box.
- The first cell holds the welcome text and a nested table to display links to tables.

Home.htm Setup – cont.

- Home page links are held in a 3x3 table, with classes defined in style.css to create the dotted lines (a repeating image) between the boxes.
- The links duplicate the functionality of menu links, and make use of #if statements to tailor the links to different users.
- Links are active over both the icon and text (i.e. “New User”).
- Which blocks of links a user sees is determined in part by the logical conditions we covered earlier, where we changed the translation.properties file to say show.<TABLE>==yes to make the table appear in the EUI.
- The second is whether the user has permission to create or edit records in the particular table. For instance, if a user can view records in the Document table but does not have ‘create’ permission, then they will see links to view Documents but not a link to submit new documents.
- For more information, please see https://wiki.agiloft.com/display/HELP/EUI+Template+Configuration.

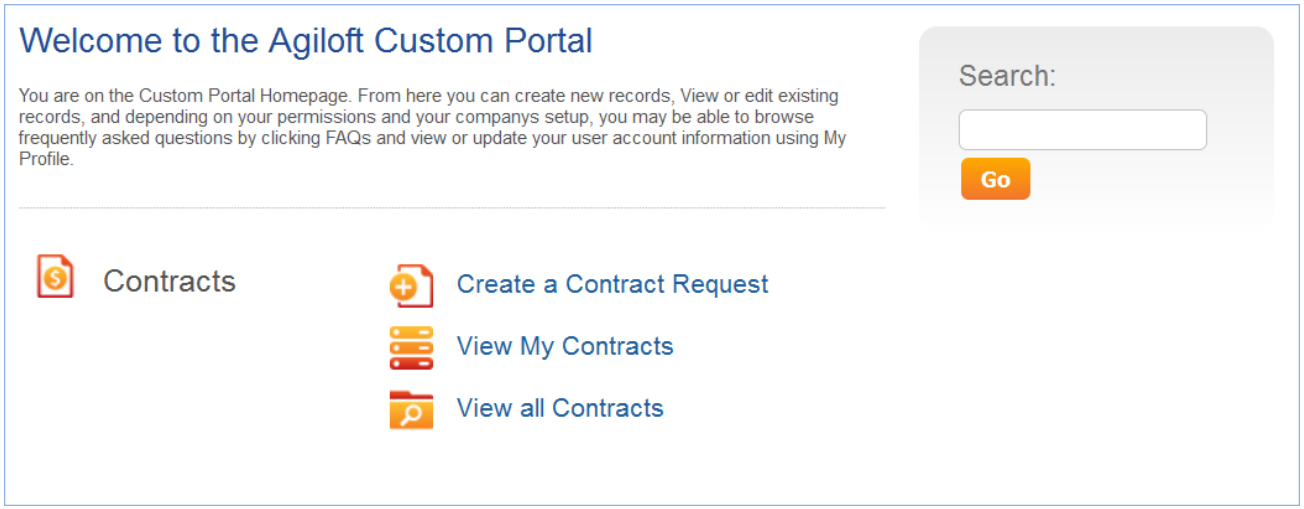
Logic of Home.htm

- All users see the My Profile menu link, except Guest group users who see a link to self register by creating a new user record.
- The following is a summary of the access granted to some of the default groups. Users in multiple groups see the superset of access granted to the individual groups.

Group name	Default access
Guest users	View documents
Document Creator	Create and view documents
Contract Creator	Create and view contracts
Approver	View contracts; View documents
Customer	Create and view support cases; View documents
Internal Customer	Create and view service requests; Create and view purchase requests; Create and view Incidents; View documents

Alternate Home Pages

- When a user only sees one block of home page links – such as the Contract Creator group – a different EUI template replaces home.htm with a special layout. This file is called home-1-table.htm. To read more about the default template configurations, please see https://wiki.agiloft.com/display/HELP/EUI+Template+Configuration.



New Page Checklist

- When creating new EUI Template record it is often best to start with a similar template and change only the elements needed to customize the contents. This will usually include:
  - EUI Template name.
  - <title> text to display in the browser's title bar.
  - Agiloft macro and its parameters.
  - <h2> page heading text.
- Each page should include the following (using the **#ew_include** macro):
  - Meta.htm (include in the <head> section).
  - Header.htm (include as first line under in default setup).
  - Footer.htm (include just before the closing tag for).
- You will also need to:
  - Add a link from menu.htm and possibly modify the menuChange function.
  - Set up group permissions and create an end user layout.

Unit Review

- In this unit you have learned:
  - How to use various macros to add functions to the EUI, including:

    #ew_create_record
    #ew_edit_record
    #ew_table
    #ew_image
    #ew_forward
    #ew_userid
    #ew_include

  - Adding and removing functions, including:

    Creating pages.
    Adding links.
    Setting up FAQs.

  - Changing the logo, background color, and conditional branding using CSS and the methods **$ewUser.isInGroup()**, **$ewUser.isInTeam()** and **$ewText.get()**.