









EUI Template Reference

The Agiloft 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 to match your branding and web design. This article contains a description for each of the main EUI templates. To see the code for each template, go to the EUI Templates table in your knowledgebase and open the desired template. For more detailed instructions on editing the templates, see [Working with Default EUI Templates](#).

Each page of the EUI has a corresponding record in the EUI Templates table. The Name field shows the full template name including the file extension, such as .htm, .html, .css, .php, or .properties.

<input type="checkbox"/>	Edit	ID ↑	Name	Date Updated
<input type="checkbox"/>		1	main.php	Mar 20 2020 06:37
<input type="checkbox"/>		2	new_servicerequest.html	May 03 2019 04:36
<input type="checkbox"/>		3	style.css	Oct 19 2020 23:15
<input type="checkbox"/>		4	footer.htm	Dec 18 2019 11:10
<input type="checkbox"/>		5	header.htm	Dec 14 2018 17:25
<input type="checkbox"/>		6	home.htm	Oct 23 2020 07:03
<input type="checkbox"/>		7	login.vm	Oct 12 2017 07:45
<input type="checkbox"/>		8	menu.htm	Oct 23 2020 06:53

EUI Templates table with Name column highlighted

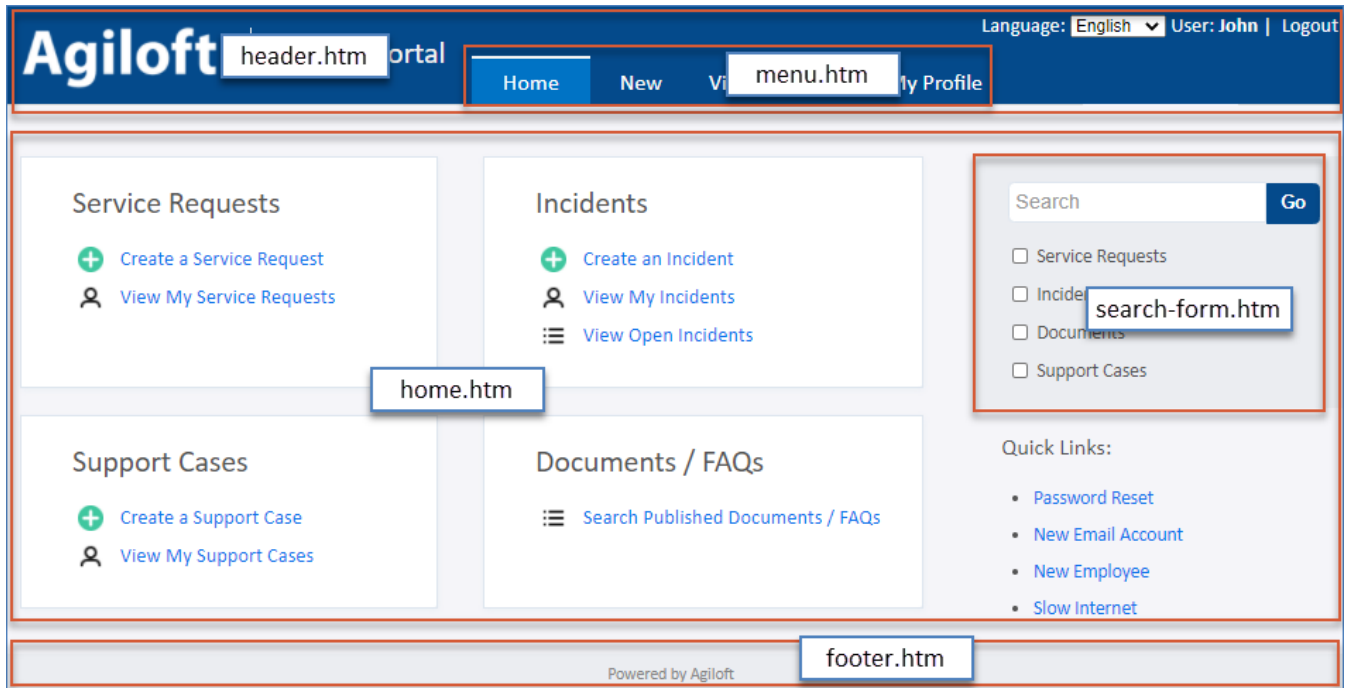
The Body field holds the code, and modifying the code makes changes to the EUI. For this reason, we recommend making a copy before editing any template records so that you can revert to the old code if necessary.

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. Here are other examples of these "helper" files:

- `iresize.js`: Contains a piece of JavaScript
- `search-form.htm`: Contains the search block found on every home page
- `meta.htm`: Holds meta data included in every page

main.php

The standard login pages provided by Agiloft are set to direct a user to `main.php` when logging into the custom portal. The following wireframe image shows how the elements contained in the `main.php` page fit together:



The `main.php` file does not directly contain the elements that are displayed to the user on the page. Instead, `main.php` uses the `#ew_include()` macro to pull in the contents of three other HTML files:

- `header.htm`
- `home.htm`
- `footer.htm`

`Header.htm` in turn includes the file `menu.htm`, and `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.

Although a login page does have to specify which page a user is taken to, it does not have to be the `main.php` page. Different login pages for the same project could even direct users to different landing pages.

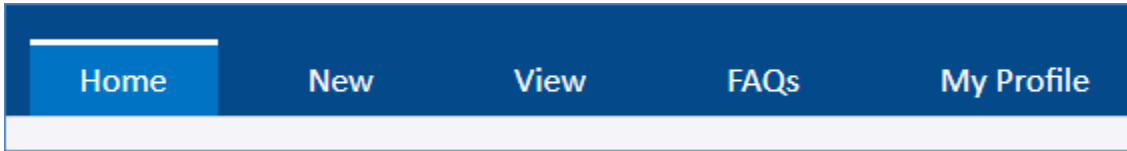
home.htm

The `home.htm` template file is the main part of the page that users see when they log in to the EUI. It makes heavy use of conditionals to show different tables to different user groups. An HTML table lines up the Agiloft tables in columns and rows. On the EUI, the horizontal and vertical lines are table cells with particular CSS classes designated. Using conditionals with this type of layout provides a relatively compact presentation when there are only a few options for each user.

You do not have to use `home.htm` if it is not suitable for your needs. In `main.php`, you can replace the reference to `home.htm` with a different EUI template to serve as your main home page. See [Working with Home Pages](#) for more information.

menu.htm

The `menu.htm` template file creates the navigation menu, or the tabs across the top of the screen.



Each tab of the menu is held in an unordered HTML list in `menu.htm`. The template file also contains a script near the bottom that highlights the current tab. If you add any extra tabs to the EUI, ensure that you give each new item a unique ID in the script.

The area used by the tabs expands to accommodate any extra tabs added. If the area used by the tabs becomes larger than the area available to the menu, it creates a second line, which generally looks bad. For this reason, we recommend that you limit the number of tabs you add, use shorter tab names, or add line breaks into the tab names so that users see a limited number of tabs. You can also create drop-down menus for the extra items. For more information on how to do this, see the [EUI Setup Tips](#) page.

header.htm

The `header.htm` template file controls the appearance of the EUI header and also references `menu.htm` to display the navigation menu. `Header.htm` contains the reference to the logo, plus user information and a logout link in the upper right corner. The `$ewText.get` method uses the `userinfo.name.label` and `userinfo.logout.link` variables to retrieve the words "User:" and "Logout," respectively, from the `translation.properties` file.

translation.properties

The `translation.properties` file is actually part of a family of files. The default Agiloft setup includes `translation.properties`, `translation_ru.properties`, and `translation_zh.properties`. These records hold the text retrieved by the `$ewText.get` method for English, Russian, and Chinese, respectively. For more information about `translation.properties`, localization, and the EUI, see [Text and Localization in the EUI](#).

style.css

The `style.css` file contains the CSS code for the EUI, which defines the page formatting and can be updated as desired. By default, `style.css` is already referenced on each page by including the `meta.htm` file, which calls `style.css` along with several other standard scripts. You can use this same shortcut on pages you create.



When editing the CSS, keep in mind that by default the `<div>` in `home.htm` have their location determined by the HTML table and do not use relative positioning. This means that the location of a particular `<div>` on the page is determined by its placement within the HTML table. If you want to update the `style.css` to use positioning, then you must remove the HTML table in `home.htm` or your positioning will not work as intended.