

# Working with the Default EUI Templates

Last Updated: June 21, 2018

[www.agiloft.com/documentation/working-with-default-eui-templates.pdf](http://www.agiloft.com/documentation/working-with-default-eui-templates.pdf)

# Contents

---

Working with the Default EUI Templates .....	3
What Controls the Visibility of Elements on Home Pages .....	3
Controlling Visibility with Permissions .....	5
Controlling Visibility with translation.properties .....	5
Home Page Layouts .....	7
Making Changes or Substituting a New Home Page .....	7
Changing the Order of Tables in the Home Page .....	7
Default Template Layouts .....	8
How Text is Displayed in the EUI .....	9
Changing Text in the Interface .....	10
Navigation Menu Structure .....	11
Changing which Menu Items a User Sees .....	11
Adding a New Table to the EUI .....	11
Code Samples .....	13
Home Page Table Block .....	13
Navigation Menu .....	15
Common Permission Parameters .....	19

# Working with the Default EUI Templates

---

The May 2018 version of the Agiloft Standard System Knowledgebase, our most popular out-of-the-box solution, offers a more dynamic, easy to customize End User Interface (EUI). This guide provides instructions on how to control which tables are displayed on the home page and menu bar, add new tables to these sections, modify the text of links, and use permissions to dynamically control the visibility of elements such as tables and links.

**Notes:** The end user interface is controlled by records in the EUI Templates table. These records contain the HTML for the pages that are displayed and control the functionality and appearance of the EUI. For more information on getting started with the EUI, please review Unit 19 of the online Administrator Training at <http://www.agiloft.com/documentation/unit-19-eui.pdf>.

The EUI uses macros and methods which are interpreted by the Apache [Velocity](#) template engine. For more information on the macros and methods used in an Agiloft EUI, please refer to the Macros Reference section of the EUI Tips and Reference Guide at <http://www.agiloft.com/documentation/eui-reference-guide.pdf>.

## What Controls the Visibility of Elements on Home Pages

---

What a specific user sees when they log in depends on two things:

- Whether the user has the right group permissions to see that table or item.
- Whether a table (e.g., Contracts or Support Cases) is “turned on” in the `translation.properties` file. If a table is not turned on, then end users will not see it on the home page, no matter what permissions they have.

To illustrate this, the three images below show the home page contents for three different users: one in the Admin group, one in the Internal Customer group, and the last in the Customer group. Each user sees different contents derived from the same home page.


















 Helpdesk Cases	 Create a Helpdesk Case  My Helpdesk Cases  FAQs
 Support Cases	 Create a Support Case  View my Support Cases  FAQs
 Contracts	 Create a Contract Request  View my Contracts  View all Contracts
 Documents	 Create a Document  View my Documents  View All Documents  FAQs

Figure 1. An Admin group member logged into the EUI sees four tables and sets of links.







 Helpdesk Cases	 Create a Helpdesk Case  My Helpdesk Cases  FAQs
 Documents	 FAQs

Figure 2. A user in the Internal Customer group sees Helpdesk Cases and Document FAQs.





 Support Cases	 Create a Support Case  View my Support Cases  FAQs
---	--

Figure 3. The user in the Customer group sees only Support Cases.

# Controlling Visibility with Permissions

When users log in, their detailed group permissions are loaded into the system; whether the user has a certain permission is used to control the visibility of elements in the EUI through use of the `$ewPermission.table()` user method.

This method requires two parameters, the logical table name and the permission parameter, in the form `$ewPermission.table("table","permission")`.

For example, `$ewPermission.table("case","view_own")` returns “true” if the logged-in user has permission to view their own records in the Support Cases table.

The `$ewPermission.table()` method can be inserted in an `#if` condition surrounding an element such as a link or menu item. These conditions are used throughout the records in the EUI Templates table to dynamically control element visibility and hide options from a user without the permission to complete that action. For example:

```
#if ($ewPermission.table("case","view_own"))
```

To change what permission is used to control the display of a particular item, edit the `$ewPermission.table` parameters to the desired permission values. Permission controls can be combined with logical operators such as an “AND” (&&) or an “OR” (||) to further filter and control what is displayed. Please review the [Code Samples](#) section for examples.

For a list of [Common Permission Parameters](#), review the section below.

For more information on Agiloft’s EUI macros, please refer to the [EUI Tips and Reference Guide](#).

## Controlling Visibility with translation.properties

The record in the EUI Templates table named `translation.properties` is where we control table visibility and all custom text shown in the interface.

There are alternate versions of this record for other languages with the same controls (e.g., `translation_ru.properties` for Russian), but we will focus on the English version here.

**Note:** If your KB is available in multiple languages, be sure to update the controls in each translation file when making changes.

## Table Visibility

Near the top of the Body field are the main visibility controls for tables and whether they are displayed in the EUI, beginning with `show.contracts=yes`. The default home page code includes the following tables: Helpdesk Cases, Support Cases, Change Requests, Contracts, Documents, Tasks, and Repair Requests.

If the `show.[table]` variable is set to “yes” for a given table, then that table is included in the interface for those users with permissions to view it. If `show.[table]` equals “no”, then the table will not be shown on the home page or menus to any users, regardless of their permissions. This offers a quick way to turn a table on or off for the EUI.

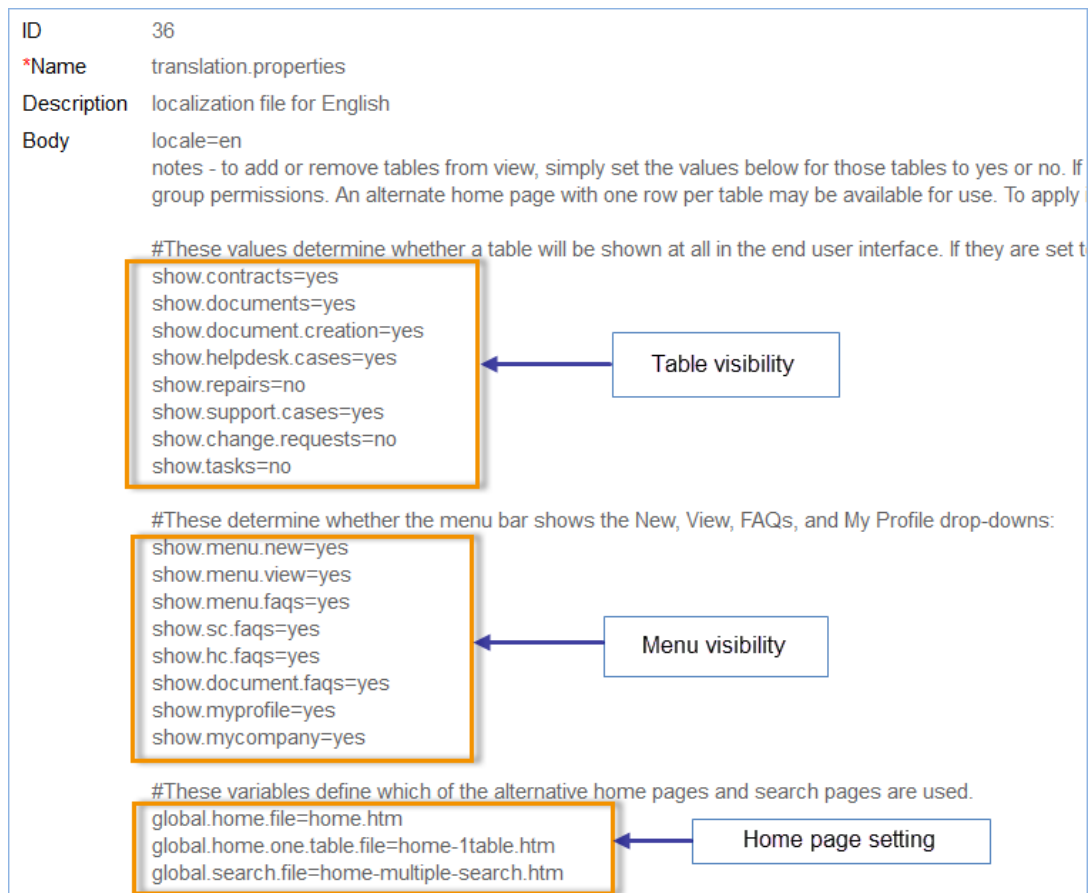


Figure 4. The top section of each `translation.properties` file includes variables to control the visibility of tables and menu items.

## Menu Item Visibility

Below the table controls is another section to control the visibility of items in the navigational menu bar, beginning with `show.menu.new=yes`. These variables define whether the New, View, FAQs, and My Profile tabs are visible, as pictured below.

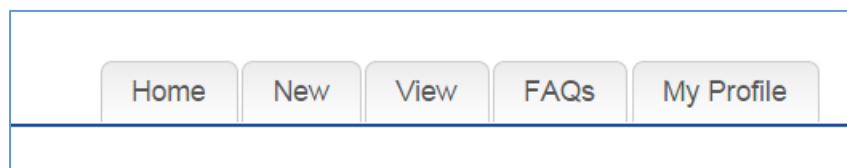


Figure 5. The EUI navigation menu.

- `show.menu.new=yes` makes the New menu tab available.
- `show.menu.view=yes` makes the View menu tab available.
- `show.menu.faqs=yes` makes the FAQs menu tab available.
- `show.myprofile=yes` makes the My Profile menu tab available.

The FAQ interface is available by default for the Helpdesk Cases, Support Cases, and Documents tables. We've included individual variables for each FAQ, e.g. `show.sc.faqs`, to provide a quick way to turn these FAQs on or off.

What actually appears in the drop-down under each menu item still depends on the user's permission to create records, view records, and view FAQs.

## Home Page Layouts

---

Near the top of the `translation.properties` file are three variables that define the home page template files.

The variable `global.home.file` defines which of two predefined home pages to use: the default value of `home.htm`, or `home-1column.htm`:

```
global.home.file=home-1column.htm      #The default home page is home.htm,
or you can use home-1column.htm by replacing this variable

global.home.one.table.file=home-1table.htm

global.search.file=home-multiple-search.htm
```

In addition to these default home pages, the code automatically substitutes a third home page if the user who logs in only has access to one table, since the other home page layouts are not optimized for displaying a single table. This third home page is defined by the variable `global.home.one.table.file` and by default it is set to the `home-1table.htm` template.

The home page code defines which tables are shown to which users, the ordering of tables, and other formatting. Of course, you may create your own version of a home page if you don't wish to use our predefined versions.

While some page components are static, such as the Logout link, other links and headings are controlled by a combination of user permissions and variables defined in `translations.properties`.

These conditions include the values of the `show.[table]` variable, as well as the user's specific permissions, returned by the `$ewPermission.table` method.

## Making Changes or Substituting a New Home Page

To make changes to the style or layout of your end user interface home page, copy and edit one of the default home page records, or create a new template record. Then, update the `global.home` and `global.home.one.table` variables in `translation.properties` to point to your new page.

## Changing the Order of Tables in the Home Page

There is a block of code for each table included in the home page; the order of these sections determines the order in which tables are displayed to users.

Therefore, to change the order of the tables, you can cut and paste one table's section above or below another one. Just be careful to include all of the section related to that table: each section should start with an `#if` and end with an `#end` (watch out for nested if-end conditions).

Specifically, each block of table code begins with:

```
#if ($ewText.get("show.[table]")=="yes")
```

Please review the [Home Page Table Block](#) example code from the home.htm template, the default home page, to see how the permission-based conditions are used.

We recommend first making a copy of the template text before making significant changes.

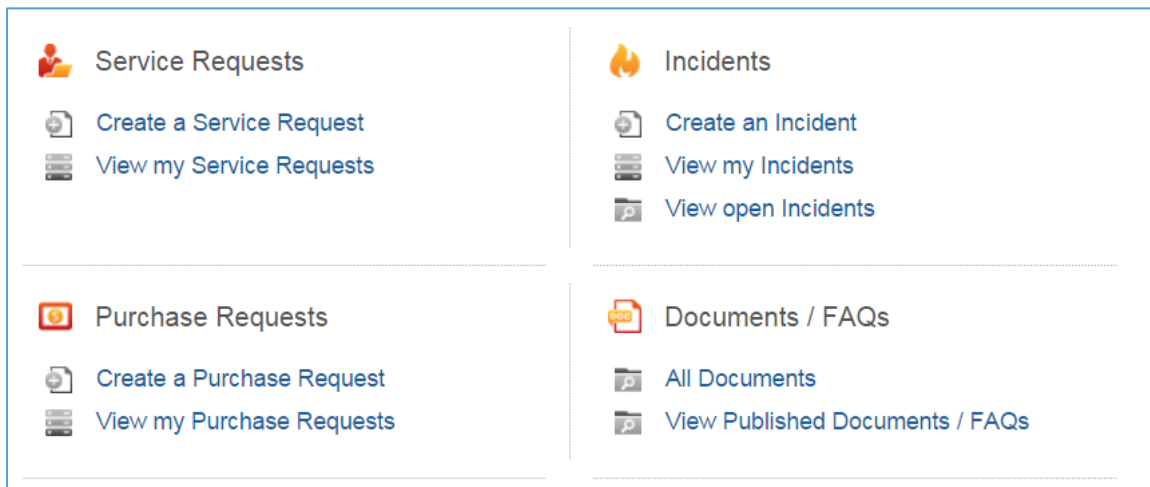
## Default Template Layouts

The default home page layouts are:

### home.htm—Default layout

The default home.htm template provides a dynamic two-column display; the system dynamically calculates the layout to list the available tables in order from left-to-right and top-down.

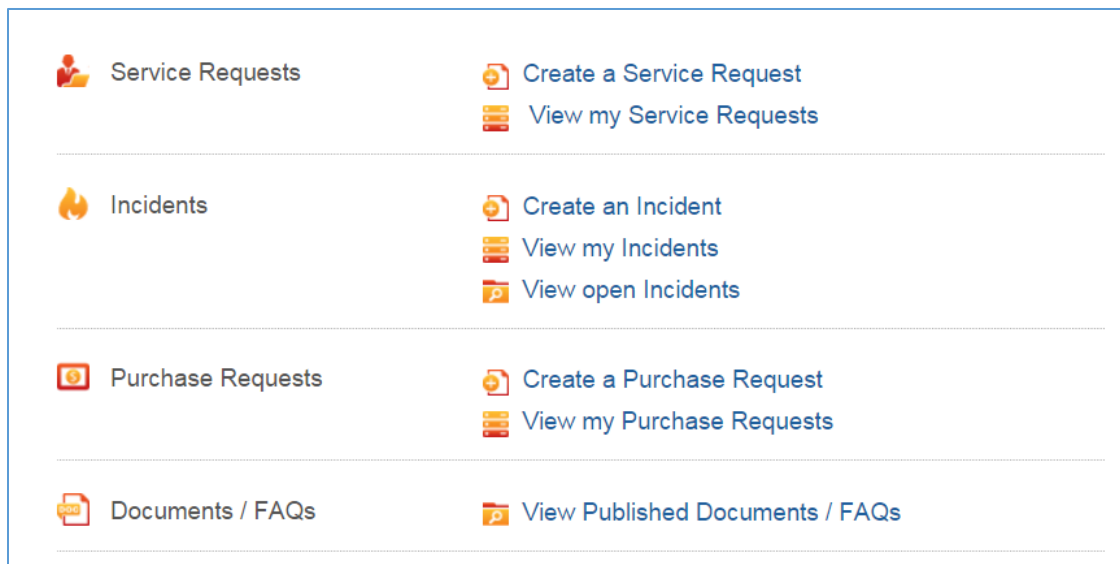
Available links to create or view items appear under each table heading, as pictured here.



### home-1column.htm—Alternative layout

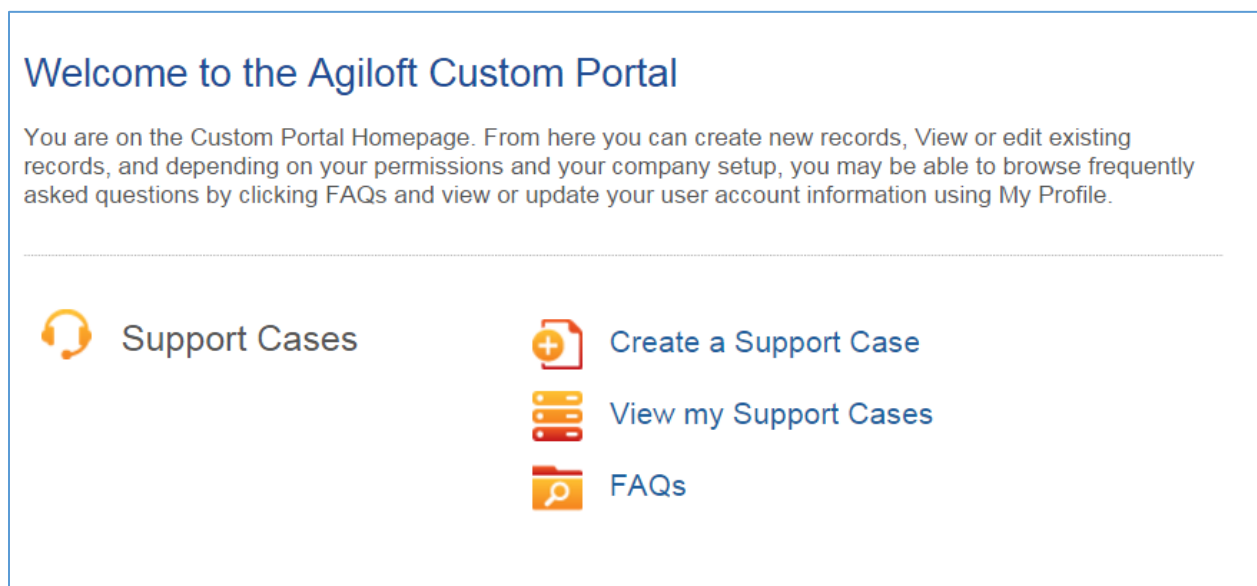
The alternative home page home-1column.htm arranges each table on its own row with the links in a second column to the right:





## home-1table.htm—Default one-table layout

The `home-1table.htm` template uses a similar layout to the `home-1column.htm` template, above. The primary differences are larger icons and text in `home-1table.htm`, and an additional block of text added to fill space on the home page.



## How Text is Displayed in the EUI

Text that appears throughout the EUI is controlled by the macro `$ewText.get()` and a set of variables in the `translation.properties` record.

The macro `$ewText.get("key")` finds the appropriate `translation.properties` file based on the user's session locale. In the `translation.properties` file, each "key" is associated with a text string, and the `$ewText.get` macro substitutes this value in the page.

For instance, the home page contains the following link:

```
<a href='#ew_forward("new_helpdeskcase.html")'>$ewText.get('home.hc.new.link')</a>
```

The link's text is defined by the `home.hc.new.link` variable in the appropriate translation file, for instance, "Create a Helpdesk Case" for English or "Создать запрос" for Russian.

In `translation.properties`, the text keys begin with `home.doc.h3`.

```
#These variables provide the text on the home pages
home.doc.h3=Documents
home.doc.faq.link=FAQs
home.doc.new.link=Create a Document
home.doc.my.link=View My Documents
home.doc.all.link=View All Documents
home.hc.h3=Helpdesk Cases
home.hc.faq.link=FAQs
home.hc.new.link=Create a Helpdesk Case
home.hc.my.link=My Helpdesk Cases
home.rep.h3=Repair Requests
home.rep.new.link=Create a Repair Request
home.rep.my.link=View My Repair Requests
home.sc.h3=Support Cases
home.sc.faq.link=FAQs
home.sc.my.link=View My Support Cases
home.sc.new.link=Create a Support Case
home.con.h3=Contracts
home.con.my.link=View My Contracts
home.con.all.link=View All Contracts
home.con.new.link=Create a Contract Request
home.cr.h3=Change Requests
```

## Changing Text in the Interface

If you want to alter the text in a template page, the process is to:

1. Open the applicable template to find the name of the variable that defines the text you want to change.
2. Open the `translation.properties` template and find the variable, then edit the text that follows the equals sign (=).

Text variables are organized by function (home page, menu, title bar) and then by table to help you determine which variables apply to which text based on the variable name.

## Why not enter text directly in page templates?

You might feel that substituting variables is a rather indirect way to manage text on the page, and indeed it would be simpler to edit text directly on the various pages. There is nothing to prevent you from bypassing the `$ewText.get` macro and simply entering text directly into the EUI Template pages if you want to.

There are two main reasons the interface uses variables for text:

- Support for multiple language usage. By using variables, the admin can create a full translation of the end user interface by simply copying the `translation.properties` template to a new one with a different language attached, e.g. `translation_fr.properties`. The text values in that file can then be translated so that someone logging in to the program with the French locale will see the translated French text values.
- Consistency and ease of changes. It is easier to make a consistent terminology change when the text variables are separated into a single file. Each home page variant references the same set of text variables. By using the same variables in each home page, if you decide to change the name of a table, for instance, from Support Cases to Support Tickets, you can go to the translation templates and make all the terminology changes across the board in one place.

## Navigation Menu Structure

---

The navigation menu and its contents are defined in the `menu.htm` EUI template. Please refer to the [Code Samples](#) section for an example of that template's text for the section controlling the New menu. You can see that it uses permission controls very similar to those used in the home page.

Note that the menu template is used to count the number of tables available to a user and if the table count in the menu results in a total value of 1, then the home page defined as the single table home page is displayed to the user.

## Changing which Menu Items a User Sees

Just as in the home page, the menu items are controlled by a combination of the `show.[table]` variable and user permissions, so if you want to hide the item from all users, the easy way is to update the `translation.properties` file to change the `show.[table]` to `no`. If it should be shown to only users with specific permissions, then the approach is to update the relevant `#if` condition to use the appropriate permissions.

## Adding a New Table to the EUI

---

If you want to add a table to the EUI that is not included in the default setup, you must update the EUI templates to provide this access.

For this example, we'll suppose you have a custom table called Invoices and you would like to add functions to the EUI such as "Create an Invoice" and "My Invoices," a link to view invoices for end users.

Below are the basic recommended steps for adding a new table:

1. Create new template pages that will be launched from the home page. We recommend copying a similar EUI template record and then changing the page-specific information such as the logical table name, macro parameters, and text variables. For instance, to create a page to display the Invoices table, copy the `my_supportcases.html` page into a new template called `my_invoices.html`; to create a "Create an Invoice" page, copy the `new_supportcase.html` template.
  - a. Edit the new template pages to update the logical table name, change the default view, add any default saved searches, and so on.
  - b. If the page you copy from includes a search block, you will need to create a corresponding subsidiary search page for the new table, Invoices.
  - c. Change the variables within the `$ewText.get` macro. Note down the variables you create and the text strings they represent. These variables and values must be added to `translation.properties` in the following step.
2. In the `translation.properties` file (and its variants for multilingual KBs):
  - a. Add the line `show.invoices=yes`  
This will let you turn the Invoices table on or off quickly and allows you to use that variable in the new section of table code. For instance, in `home.htm` or `menu.htm` where you previously had `#if ($ewText.get("show.support.cases")== "yes"`, replace `show.support.cases` with `show.invoices`.
  - b. Add any text variables created for the subsidiary pages in the previous step.
  - c. Create new text variables for the home page and menu: the table title, create link, view links, menu tabs, etc. Use the set of variable codes from an existing table and create new variables using a similar syntax. Since we are copying from Support Cases in this example, copy the text variables (in the left column) and create new variables for the Invoices table in a similar syntax (shown in the right-hand column).

Support Case Text	Invoice Text
<code>home.sc.h3=Support Cases</code>	<code>home.invoice.h3=Support Cases</code>
<code>home.sc.my.link=View My Support Cases</code>	<code>home.invoice.my.link=View My Invoices</code>
<code>home.sc.new.link=Create a Support Case</code>	<code>home.invoice.new.link=Create an Invoice</code>
<code>menu.new.sc=Support Case</code>	<code>menu.new.invoice=Invoice</code>
<code>menu.view.sc.my=My Support Cases</code>	<code>menu.view.invoice.my=My Invoices</code>

3. Edit the default home page(s) you are using.

- a. Copy an existing table's complete section of code, e.g. the Support Case section, and paste it to create a new section of the home page template. Paste the section into the appropriate position so the tables are ordered as you prefer.
- b. Modify the copied block of code by replacing variables and parameters with new values associated with the table you want to display (Invoices). For example:
  - i. Update the #ew\_forward macro to point to the new template pages created in the previous step.
  - ii. Update the \$ewText.get statements to use the variables created in the previous step.
  - iii. Update the logical table name (i.e., case) wherever it appeared.
4. To add links from the menu, add new sections to the menu.htm template in the appropriate places (e.g., under the New and View sections). Copy an existing block of code and change the table name, text variables, and criteria as needed.
5. Of course, you must set group permissions to give users the appropriate access to the Invoices table and its records.

## Code Samples

---

### Home Page Table Block

Below is an example section of code for a single table, the Support Cases table, as defined in the default home.htm template. Comments are in blue, and visibility controls are highlighted in yellow.



```
#if ($ewText.get("show.support.cases")=="yes" &&
($ewPermission.table("case","view_own") ||
$ewPermission.table("case","v_own_with_ssearch") ||
$ewPermission.table("case","view faq") || $ewPermission.table("case","view
faq_with_ss")))

#set($rowcount=$rowcount+1)
<td class="home-section">
```



If the show support case FAQs is yes and the user has View FAQs or view FAQs with saved search permissions on the Support Case table, show the FAQs link.

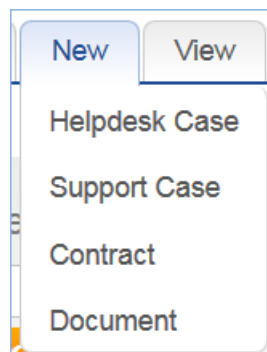
```
</div>
</div>
</td>
  #if ($rowcount%2==0)
</tr>
<tr>
  <td class="h-line"></td>
  <td></td>
  <td class="h-line"></td>
</tr>
<tr>
  #else
  <td class="v-line"></td>
  #end
#end
```

This defines whether it is the first column or second column and inserts the appropriate vertical or horizontal lines.

## Navigation Menu

### ‘New’ Menu

Below is the section of menu.htm that creates the New menu drop-down in the EUI.



```

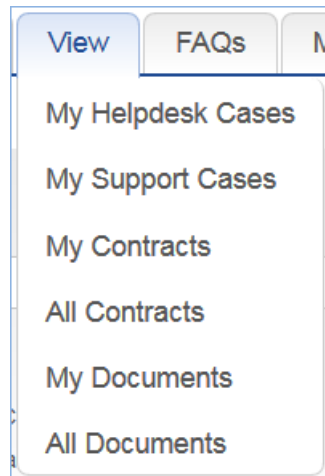
# if ($ewText.get("show.menu.new")=="yes")
    <li id="new" class="submenu"><a>$ewText.get("menu.new")</a>
        <ul>
# if ($ewText.get("show.helpdesk.cases")=="yes" &&
$ewPermission.table("helpdesk_case","create"))
            <li><a
href='#ew_forward("new_helpdeskcase.html")'>$ewText.get("menu.new.hc")</a></li>
# end
# if ($ewText.get("show.support.cases")=="yes" &&
$ewPermission.table("case","create"))
            <li><a
href='#ew_forward("new_supportcase.html")'>$ewText.get("menu.new.sc")</a></li>
# end
# if ($ewText.get("show.change.requests")=="yes" &&
$ewPermission.table("change_request","create"))
            <li><a
href='#ew_forward("new_changerequest.html")'>$ewText.get("menu.new.cr")</a></li>
# end
# if ($ewText.get("show.contracts")=="yes" &&
$ewPermission.table("contract","create"))
            <li><a
href='#ew_forward("new_contract.html")'>$ewText.get("menu.new.con")</a></li>
# end
# if ($ewText.get("show.documents")=="yes" &&
$ewText.get("show.document.creation")=="yes" &&
$ewPermission.table("document","create"))
            <li><a
href='#ew_forward("new_document.html")'>$ewText.get("menu.new.doc")</a></li>
# end
# if ($ewText.get("show.tasks")=="yes" &&
$ewPermission.table("activities","create"))
            <li><a
href='#ew_forward("new_task.html")'>$ewText.get("menu.new.task")</a></li>
# end
# if ($ewText.get("show.repairs")=="yes" &&
$ewPermission.table("repair_request","create"))
            <li><a
href='#ew_forward("new_repair.html")'>$ewText.get("menu.new.rep")</a></li>
# end
Essentially the items are controlled by the show.[table]=yes variable and the
create permissions for a particular user.

```



## ‘View’ Menu

Below is the section of menu.htm that creates the View menu drop-down in the EUI.



```
#if ($ewText.get("show.menu.view")=="yes")
    <li id="view" class="submenu"><a>$ewText.get("menu.view")</a>
        <ul>
            #if ($ewText.get("show.helpdesk.cases")=="yes" &&
                ($ewPermission.table("helpdesk_case","view_own") ||
                $ewPermission.table("helpdesk_case","v_own_with_ssearch")))
                #set($htable=1)
                #set($tablecount=$tablecount+1)
                <li><a
href='#ew_forward("my_helpdeskcases.html")'>$ewText.get("menu.view.hc.my")</a>
                </li>
            #end
            #if ($ewText.get("show.support.cases")=="yes" &&
                ($ewPermission.table("case","view_own") ||
                $ewPermission.table("case","v_own_with_ssearch")))
                #set($sctable=1)
                #set($tablecount=$tablecount+1)
                <li><a
href='#ew_forward("my_supportcases.html")'>$ewText.get("menu.view.sc.my")</a>
                </li>
            #end
            #if ($ewText.get("show.change.requests")=="yes" &&
                ($ewPermission.table("change_request","view_own") ||
                $ewPermission.table("change_request","v_own_with_ssearch")))
                #set($tablecount=$tablecount+1)
```

```

                <li><a
href='#ew_forward("my_changerequests.html") '>$ewText.get("menu.view.cr.my")</a></li>

#end

#if ($ewText.get("show.contracts")== "yes" &&
($ewPermission.table("contract","view_own") ||
$ewPermission.table("contract","v_own_with_ssearch")))

                #set($tablecount=$tablecount+1)

                <li><a
href='#ew_forward("my_contracts.html") '>$ewText.get("menu.view.con.my")</a></li>

                <li><a
href='#ew_forward("all_contracts.html") '>$ewText.get("menu.view.con.all")</a></li>

#end

#if ($ewText.get("show.documents")== "yes" &&
($ewPermission.table("document","view_own") ||
$ewPermission.table("document","v_own_with_ssearch")))

                #set($tablecount=$tablecount+1)

                #set($doctable=1)

                <li><a
href='#ew_forward("my_documents.html") '>$ewText.get("menu.view.doc.my")</a></li>

                <li><a
href='#ew_forward("all_documents.html") '>$ewText.get("menu.view.doc.all")</a></li>

#end

#if ($ewText.get("show.tasks")== "yes" &&
($ewPermission.table("activities","view_own") ||
$ewPermission.table("activities","v_own_with_ssearch")))

                #set($tablecount=$tablecount+1)

                <li><a
href='#ew_forward("my_opentasks.html") '>$ewText.get("menu.view.opentask.my")</a></li>

#end

#if ($ewText.get("show.repairs")== "yes" &&
($ewPermission.table("repair_request","view_own") ||
$ewPermission.table("repair_request","v_own_with_ssearch")))

                #set($tablecount=$tablecount+1)

                <li><a
href='#ew_forward("my_repairs.html") '>$ewText.get("menu.view.rep.my")</a></li>
>

#end

```

# Common Permission Parameters

The table below lists common permission-based parameters for use with the `$ewPermission.table()` method. Each of these corresponds to a permission setting in the Manage Groups wizard.

Parameter	Permission
create	Create records in the table
edit_own	Edit their own records in the table
e_own_with_ssearch	Edit only their own records matching a saved search
delete_own	Delete their own records in the table
view_own	View their own records in the table
v_own_with_ssearch	View only their own records matching a saved search
view_others	View other people's records in the table
v_oth_with_ssearch	View only others' records matching a saved search
view_faq	View FAQs from the table
view_faq_with_ss	View only FAQs matching a saved search

The screenshot displays the 'Manage Groups' wizard interface, showing four panels of permissions for 'Support Cases'.

- Top Left Panel:**
  - Allow this group to:
    - ☒ Create Support Cases
      - ☐ Import multiple Support Cases from a file
      - ☐ Copy Support Cases
- Bottom Left Panel:**
  - Allow this group to:
    - ☒ Edit their own Support Cases
      - ☒ All
      - ☐ Only those that satisfy the saved search:

All Cases, ordered by Priority Edit Create New Search
    - ☐ Mass Edit multiple records in Support Case in a single operation
    - ☐ Quick Edit records from the Table View
    - ☐ Link multiple records using the Link menu
    - ☒ Print records using the printer icon
- Top Right Panel:**
  - Allow this group to:
    - ☐ Delete their Support Cases
      - ☐ Mass delete multiple Support Cases in a single operation
- Bottom Right Panel:**
  - Allow this group to:
    - ☒ View their own Support Cases
      - ☒ All
      - ☐ Only those that satisfy the saved search:

All Cases, ordered by Priority Edit Create New Search
    - ☐ View other people's Support Cases
      - ☐ All
      - ☒ Only those that satisfy the saved search:

All Cases, ordered by Priority Edit Create New Search
    - ☒ View FAQs for the Support Cases
      - ☒ All
      - ☐ Only those that satisfy the saved search:

All Cases, ordered by Priority Edit Create New Search
    - ☐ Export multiple Support Cases to a file
    - ☐ Show conversion button on menu action bar
    - ☒ Allow interaction with conversion rules