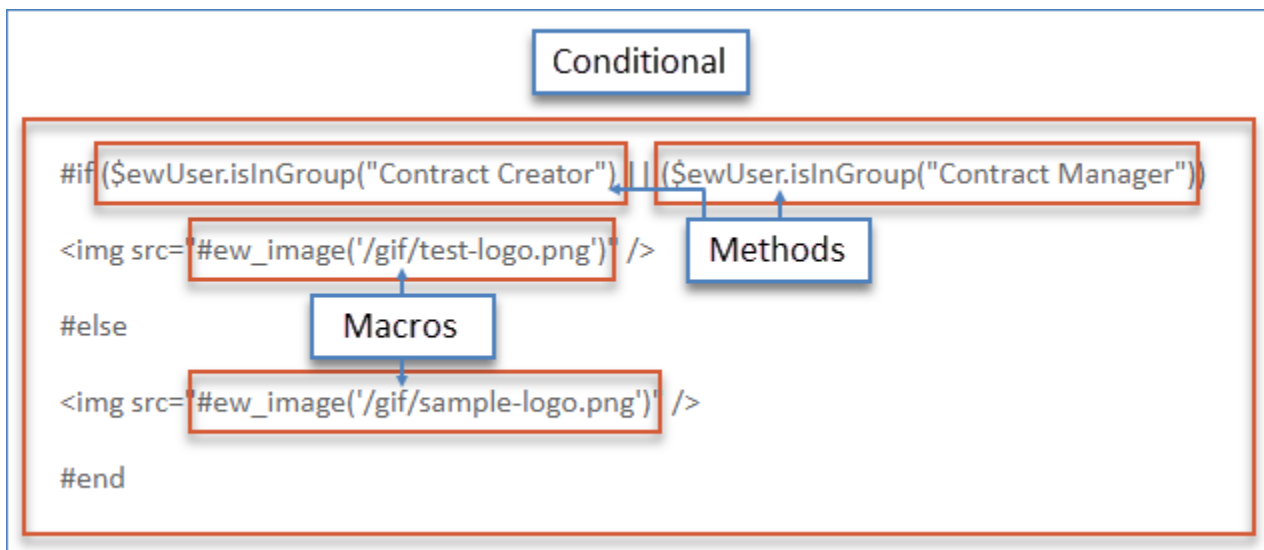


Macros, Methods, and Conditionals

Macros are short, reusable pieces of code that represent a longer set of instructions for the software or browser to run. Macros allow developers to save time and simplify their code. In the Agiloft EUI, macros are shorthand instructions for linking to pages, requesting data from the system, creating table views, and other operations.

Macros are often combined with methods and conditional statements, or conditionals, to control the visibility of a macro's content. Methods typically function as the actual condition of a conditional, and the conditional determines what the user sees on the EUI. For example, consider two macros that each display an image. A conditional might tell the system to use one macro to show image A if the user is in group A and use the other macro to display image B if the user is in group B. In this case, the method checks the user's group, and the conditional is the entire statement that uses the user's group membership to determine which image is displayed.



Understanding how macros, methods, and conditionals work in the EUI is helpful if you plan to make any customizations to the interface. Each of these elements is an essential part to how the EUI functions, so knowing how they work can help you create an EUI that suits your needs.

Macros

Agiloft uses the Velocity Template Language (VTL) template engine to process macros, which uses a special syntax:

```
#macro_name(" $param1" " $param2" " $param3" ...)
```

Macros begin with a "#" sign followed by the name of the macro, and many macros also require one or more parameters, which function as additional instructions. Parameters immediately follow the macro name and are enclosed in parentheses. Parameters are each further enclosed in single or double quotes, depending on context. Although the actual name of a parameter includes a dollar sign, in practice the parameter does not typically use a dollar sign when it's replaced by a value.

Macros with Multiple Parameters

When a macro uses multiple parameters, the order of the parameters is important. The system uses the parameter order to determine which parameter you're setting. Even if you're not setting a specific parameter, or if you're leaving it at its default value, you need to leave a space for it in the code so that you retain the correct parameter order. Parameters that aren't set or that are left at their default value are indicated with a pair of empty quotation marks.

In practice, a macro with several parameters may look like this:

```
#ew_table("contract" "" $defaultsearch ""  
"showToolBar=false&showNavigation=true&showfastsearch=false" "my_table" "")
```

This macro contains seven parameters. One parameter, `$defaultsearch`, is a special variable parameter that references a saved search and uses a dollar sign but doesn't require quotation marks. See [Using Default Searches](#) for more information on how this parameter works. You can find the other six parameters by counting the pairs of quotation marks. Notice that three parameters are empty, meaning that they use a default value. For more information on this macro and its parameters, as well as other macros, see [EUI Macro Reference](#).

Macros and Page Types

In the default EUI setup, each of the basic page types corresponds to an Agiloft macro that uses parameters to reference the table, record, or form. The macros can also apply search filters or set additional parameters.

- Table views, such as the My Support Cases or All Documents pages, use the `#ew_table` macro, which creates an iframe and displays an Agiloft table inside it.
- Record editing pages, such as the My Profile page, use the `#ew_edit_record` macro.
- Submission forms, such as the Create a Support Case page, use the `#ew_create_record` macro.
- FAQ pages, such as the Document / FAQs page, use the `#ew_faq` macro.

For more information on these macros, see [EUI Macro Reference](#).

Conditionals and Methods

In addition to specialized macros, the Velocity Template Language (VTL) also has prebuilt directives for creating conditional statements, or conditionals. You can use conditionals within the body of an EUI template to show or hide HTML elements and macros. Conditionals are typically combined with methods, which are ways of checking the conditions, making decisions, and retrieving information. Our discussion only covers the basics, so reference the VTL user manual if you need more information.

Using Conditionals

Before including methods, let's first consider conditionals on their own:

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

This conditional displays the text "Welcome!" only if the condition is true; otherwise, nothing happens. Conditionals start with the `#if` directive and end with the `#end` directive, like in this example. However, they can also be combined with `#else` and `#elseif` directives, as well as logical operators, to create more complex conditionals. For instance, consider this example:

```
#if([condition A is true] && [condition B is true])
<p>Welcome!</p>
#else
<p>Hello!</p>
#end
```

This conditional displays the text "Welcome!" if both the conditions are true. Otherwise, it displays the text "Hello!" Similarly, consider this example:

```
#if([condition A is true] || [condition B is true])
<p>Welcome!</p>
#else
<p>Hello!</p>
#end
```

This conditional displays the same "Welcome!" text, but this time the text is displayed if either condition is true. If neither condition is true, it displays the text "Hello!"

Logical Operators

These are the logical operators you can use when comparing conditions within a conditional:

Operator	Definition
	Or
&&	And
!	Not
<	Less than

>	Greater than
==	Is equal to

Using Methods

Methods often function as the actual condition of a conditional. For instance, some common methods check a user's group membership, team membership, and permission settings, and the conditional uses the result to determine what the user sees. Methods begin with a dollar sign and, like macros, contain one or more parameters. Parameters immediately follow the method name and are enclosed in parentheses, and each parameter is further enclosed in double quotes and separated by commas. For instance, a method might look like this:

```
$ewUser.isInGroup("group name")
```

This method checks the current user's group membership to see if they're a member of the specified group. Notice that this method only has a single parameter, which contains the group name. When a method is used with a conditional and a macro, it might look like this:

```
#if ($ewUser.isInGroup("Contract Creator") || $ewUser.isInGroup("Contract Manager"))  
  
#else  
  
#end
```

This conditional uses the method to check if the current user is in the Contract Creator or Contract Manager groups. If they are, the system uses some HTML with a macro to display `contract-team-logo.png`. If the user is not in either of these groups, the system displays `sample-logo.png`.

For a list of common methods used in the EUI, see [EUI Method Reference](#).