EUI Macro Reference

This topic lists the custom macros and methods you can use in Agiloft EUI Templates.

Macros

Macros are based on the Velocity template language. Each of these can be used to insert Agiloft forms and data into the end user portal. Read more about creating a custom portal in the EUI Portal Tutorial or check out the online training.

The `$subtypeName` parameter is used in several EUI macros. It refers to the logical table name, which can be found in the Table wizard. For example, the logical table name for the Contracts table is `contract`. For subtables, the syntax is `table.subtype`, for example `person.employee` or `ticket.enhancement`.

- ew_chat
- ew_create_record
- ew_dashboard
- ew_edit_record
- ew_faq
- ew_forward
- ew_image
- ew_include
- ew_languages
- ew_languages_all
- ew_locale
- ew_logins
- ew_logout
- ew_new_search
- ewquery
- ew_recordfield
- ew_repchart
- ew_scripts
- ew_searches_list
- ew_session
- ew_status
- ew_table
- ew_user
- ew_userfield
- ew_userid
- ew_url
- ew_view_record
    - $ewPermission.table

ew_chat

```
#ew_chat($sessionHash)
```

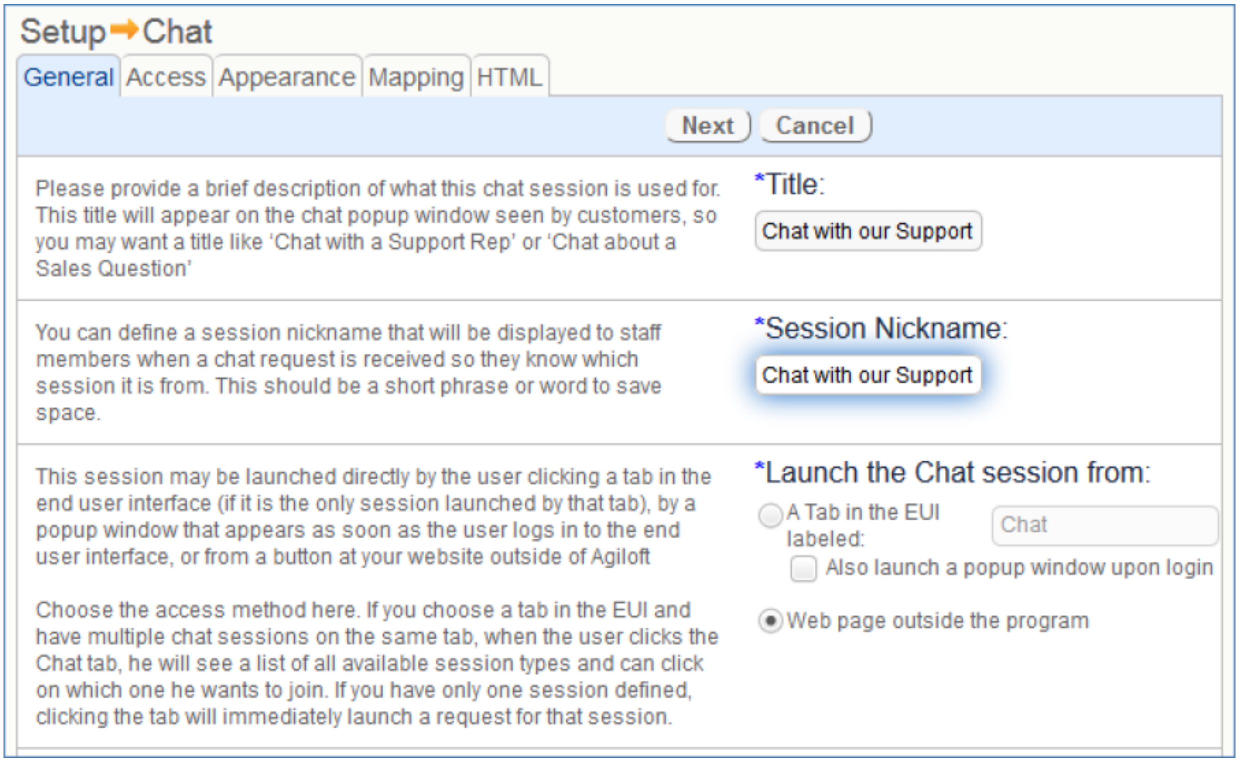
This macro lets end users start a chat session. It forms a URL with the chat session ID encoded.

Usage example:

```
<a href='#ew_chat("46326e7a79564f5a3271376f4257786b5945344e46413d3d")'
target=blank>Do chat!</a>
```

You can obtain the sessionHash from the Chat setup as follows:

1. First select "Web page outside the program" on the General tab.



Please provide a brief description of what this chat session is used for. This title will appear on the chat popup window seen by customers, so you may want a title like 'Chat with a Support Rep' or 'Chat about a Sales Question'

*Title: Chat with our Support

You can define a session nickname that will be displayed to staff members when a chat request is received so they know which session it is from. This should be a short phrase or word to save space.

*Session Nickname: Chat with our Support

This session may be launched directly by the user clicking a tab in the end user interface (if it is the only session launched by that tab), by a popup window that appears as soon as the user logs in to the end user interface, or from a button at your website outside of Agiloft

Choose the access method here. If you choose a tab in the EUI and have multiple chat sessions on the same tab, when the user clicks the Chat tab, he will see a list of all available session types and can click on which one he wants to join. If you have only one session defined, clicking the tab will immediately launch a request for that session.

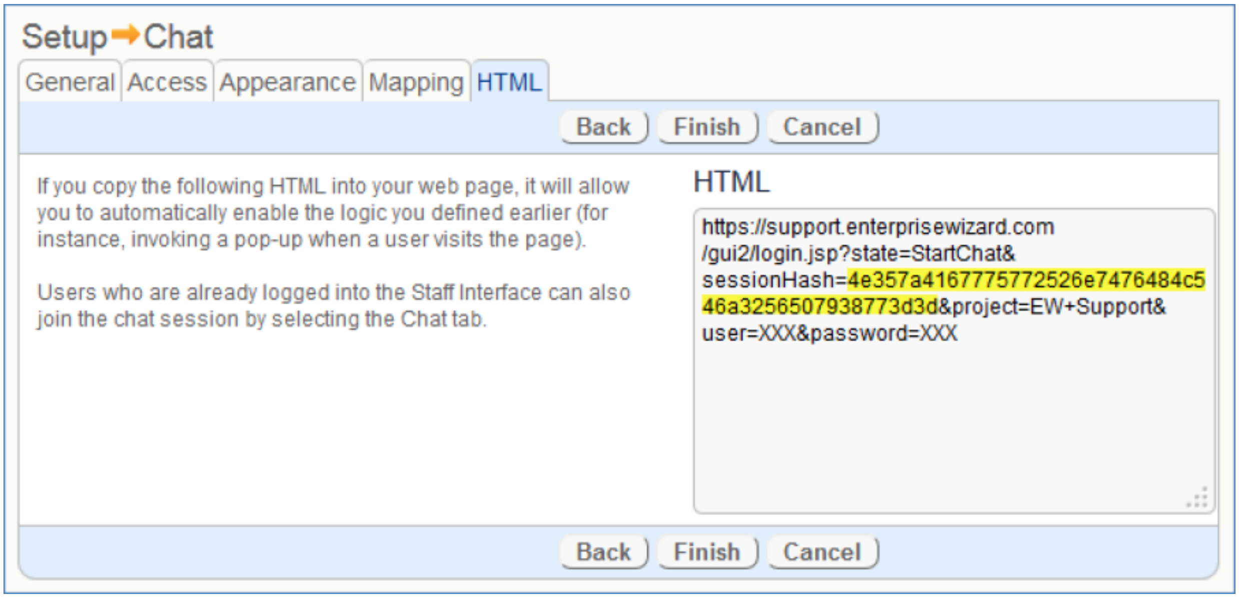
*Launch the Chat session from:

○ A Tab in the EUI labeled: Chat

□ Also launch a popup window upon login

● Web page outside the program

2. Then copy the session Hash displayed on the HTML tab.



If you copy the following HTML into your web page, it will allow you to automatically enable the logic you defined earlier (for instance, invoking a pop-up when a user visits the page).

Users who are already logged into the Staff Interface can also join the chat session by selecting the Chat tab.

HTML

```
https://support.enterprisewizard.com
/gui2/login.jsp?state=StartChat&
sessionHash=4e357a4167775772526e7476484c5
46a3256507938773d3d&project=EW+Support&
user=XXX&password=XXX
```

# ew_create_record

`#ew_create_record($subtypeName $returnURL $returnFrame)`

Use this macro to insert a new record form into an EUI template. It creates a URL that links to the standard record create page for a given table.

Parameter Name	Description
`$subtypeName`	Logical name of the table in which the new record should be created. The logical name is shown in the tables wizard. Can contain subtable chains like case.question, contact.employee etc.
`$returnURL`	URL to return to after the ticket is created or cancel is pressed.
`$returnFrame`	Name of the iframe that contains the view of the table that needs to be refreshed once the operation is completed.

# ew_dashboard

`#ew_dashboard($dashboardName $showMenu)`

Use this macro to insert a dashboard URL into an EUI template. Place the URL in a frame or other HTML content container to show a dashboard in the EUI. Note that dashboards are shown according to user permissions, and that users who are not part of a power user group can only be shown Hotlink, Table, URL, Image and Text widgets.

Parameter Name	Description
`$dashboardName`	The name of the dashboard as it appears in the dashboard list. If the $dashboardName value is empty, the most recently selected dashboard is shown; if no dashboard has been selected, the default dashboard is shown.
`$showMenu`	Use $showMenu to specify whether to show the dashboards drop-down list at the top. If the $showMenu value is empty, the default setting is True, which shows the list.

ew_edit_record

```
#ew_edit_record($subtypeName $recordId $returnURL $returnFrame)
```

This macro inserts a record edit form into an EUI template. It creates a URL that links to the standard record editing page for a given table and record ID.

Parameter Name	Description
`$subtypeName`	Logical name of the table in which the new record should be created. The logical name is shown in the tables wizard. Can contain subtable chains like case.question, contact. employee etc.
`$recordId`	The ID of the record to edit
`$returnURL`	URL to return to after the ticket is created or cancel is pressed
`$returnFrame`	Name of the iframe that contains the view of the table that needs to be refreshed once the operation is completed.

ew_faq

```
#ew_faq($subtypeName)
```

Inserts the FAQ interface for a given table, as defined in the FAQ setup. See Search FAQ Setup for more information.

ew_forward

`$ew_forward($templateName)`

This URL simply creates links between EUI template pages without losing the user's session ID. Use this whenever you create links between pages; it creates a URL to forward the user to the specified template.

Parameter Name	**Description**
`$templateName`	Template name. If the template doesn't exist, using this URL will result in an error to allow the designer to debug and fix errors

Example usage:

```
<a href="#ew_forward("testContacts.htm")">Contacts</a>
```

ew_image

`#ew_image($imageName)`

Forms the URL to the image with the given name. As an admin user, you can upload images to your KB via the power user interface in the Look & Feel section. The `$imageName` parameter should be a relative path to the image from the Agiloft file directory. For example:

```
<img src="#ew_image('/gif/icon21.png')" alt="Next" />
```

Image names with spaces or special characters should be properly URL encoded. For example, linking to an image called `icon 21.png` requires the following format, replacing the space with `%20`:

```
<img src="#ew_image('/gif/icon%2021.png')" alt="Next" />
```

# ew_include

`#ew_include($templateName)`

This macro is a shortcut to include the contents of one EUI Template in another.

Parameter Name	**Description**

`$templateName`	Includes $templateName into the current one. This allows admins to reuse common pieces of code in different templates. See EUI Template Configuration for an example of how this can be used.

ew_languages

```
#ew_languages($controlName $onChange $controlStyle)
```

Renders a list of available languages for the current user with a particular HTML style. The style needs to be passed here as a parameter, because Agiloft will render the HTML control automatically, making it impossible to customize its appearance any other way.

Parameter Name	Description
`$controlName`	Required name of the HTML select list.
`$onChange`	JavaScript code to be run when a language is selected.
`$controlStyle`	CSS class name for the select list control.

ew_languages_all

```
#ew_languages_all($controlName $onChange $controlStyle)
```

Renders a list of all languages available in the system with a particular HTML style. You need to pass the style here as a parameter, because Agiloft will render the HTML control automatically, making it impossible to customize its appearance any other way.

Parameter Name	Description
`$controlName`	Required name of the HTML select list.
`$onChange`	JavaScript code to be run when a language is selected.
`$controlStyle`	CSS class name for the select list control.

ew_locale

```
#ew_locale($localeName)
```

Sets the language (locale) for the current user session to the specified one if the language is available for the user.

Parameter Name	Description
`$localeName`	ISO-639 language code ("en", "de", "fr")

ew_logins

```
#ew_login($kbName $redirectTo)
```

Prints the Agiloft login URL. This tag must be enclosed in <form> with input controls "login", "password" inside. If no user and login is passed, then system will attempt to login under the predefined user eui2/qwerty. Create this user in your KB to use this default.

Parameter Name	Description
`$kbName`	name of KB to login
`$redirectTo`	name of template to be used after successful login

ew_logout

```
#ew_logout($url)
```

Logs out of Agiloft and redirects the user to the specified URL.

Parameter Name	Description
`$url`	URL to be redirected after logout, such as a confirmation message or intranet home page.

Example usage

```
<a href="#ew_logout("http://www.example.com")">Click to logout</a>
```

ew_new_search

```
#ew_new_search($subtypeName $frameName $allowSaving $returnURL $params)
```

Provides a piece of JavaScript, mostly for use in onClick event handlers, that runs the new Saved Search wizard. When returning to `$returnURL`, an additional request parameter is appended for the created saved search - searchID. You can use it in your own code, for example to update a list of available Saved Searches.

Parameter Name	Description
`$subtypeName`	Table for which the search should be created
`$frameName`	The name of HTML iframe in which to display the results after the Wizard has finished.
`$allowSaving`	"true" if the user should be allowed to save the created search, otherwise "false".
`$returnURL`	URL that should be applied to $frameName after wizard has finished. If "", than the standard table search will be used.
`$params`	Visualization Parameters formed as a request query, e.g. "showstatus=false&showpages=false..." etc. They will be applied to the table view for $subtypeName. Use this parameter if you leave $returnURL empty. Otherwise add required Parameters to $returnURL directly.

ewquery

```
#ew_query($subtypeName $frameName $controlId $params)
```

Provides a piece of JavaScript, mostly for use in onClick or onChange event handlers, that runs a search for a table in an iframe.

Parameter Name	Description
`$subtypeName`	The name of the table to show
`$frameName`	The name of HTML iframe in which to display the results.
`$controlId`	The id of the HTML control which contains the query to use (for example a form field)
`$params`	String of request Parameters that will be passed to the table in $frameName. See the #ew_table macro, and the following parameter list for details.

ew_recordfield

```
#ew_recordfield ($fieldName $subtypeName $recordID)
```

Forms the URL to show the report or chart with the given name.

Parameter Name	Description
`$fieldName`	Field name in the selected table, for example "description" or "attached_files".
`$subtypeName`	Logical table name containing the record, e.g. "document" or "person.employee".
`$recordID`	The ID of the record you want to access, e.g. 224.

Example Usage

```
#ew_recordfield("description_text" "document" 224)
```

ew_repchart

```
#ew_repchart ($repchartName $repchartType $tableName)
```

Forms the URL to show the Report or Chart with the given name from the specified table.

Parameter Name	Description
`$repchartName`	Name of report or chart as listed in the Table Charts/Reports list.
`$repchartType`	Type of the report or chart. Allowed values are [html, image, excel, plain/text].
`$tableName`	Name of the table containing the report or chart.

ew_scripts

```
#ew_scripts()
```

Introduces a link to certain required JavaScript functions. The default Agiloft EUI requires some custom JavaScript code. Use it within a page's <head> tag if you are getting strange behavior and / or JavaScript errors about null and undefined objects in your portal templates.

ew_searches_list

```
#ew_searches_list($subtypeName $frameName $controlName $controlStyle $params
$selectedSearch)
```

Draws an HTML select list control with the Saved Searches available, taking into account access permissions.

Parameter Name	Description
`$subtypeName`	Name of the table for which to display the appropriate searches.
`$frameName`	Name of the iframe that contains the view of the table that needs to be refreshed once a search is selected.
`$controlName`	Required name of the HTML select list.
`$controlStyle`	CSS class name to be applied to the select list.
`$params`	String of the request Parameters that will be passed to the table in $frameName. See the #ew_table macro, and the following parameter list for details.
`$selectedSearch`	Name of the search to be selected in list initially

ew_session

```
#ew_session()
```

Prints Agiloft session ID to screen, and requires no parameters. This may be used to 'manually' construct a URL that contains the session information.

Example

```
 <a href="/gui2/eui2template/testContacts.htm;#ew_session()">Contacts</a>
```

results in:

Output

```
<a href="/gui2/eui2template/testContacts.htm;jsessionid=D7988DF0E8E173...
```

ew_status

```
#ew_status()
```

Prints to screen last operation status stored in the request, ActionStatus. This macro requires no parameters.

ew_table

```
#ew_table($subtypeName $viewName $searchName $queryString $params $frameName
$frameStyle $showLabel)
```

Draws an HTML iframe with the specified Agiloft table displayed inside it.

Parameter Name	Description
`$subtypeName`	The table to show.
`$viewName`	The view to use for table. Note that it's localized for now, so different names should be used for different languages
`$searchName`	The search name to apply to the table. Can be used simultaneously with $queryString
`$queryString`	The query to be applied to the table. When used, the result is the same as if the user had typed the contents of this string into the search box. See the #ew_query macro below.
`$params`	Visualization Parameters formed as a request query, e.g. "showstatus=false&showpages=false..." etc. Use it for precise tuning of the table's display features. See the parameter list at the end of this section for more details.
`$frameName`	The required name of the resulting iframe. This is needed so that you can refer to it in other macros or pieces of JavaScript (for example to refresh the table view after a search has been applied - see the tutorial).
`$frameStyle`	The CSS style name to be applied to the iframe.

ew_user

```
#ew_user()
```

Prints the current user login to screen. This macro does not take parameters.

ew_userfield

```
#ew_userfield($fieldName)
```

Prints the current user's record field value. This macro is rendered to be human-readable; for example, choice values will be returned as 'Yes'.

Parameter Name	Description
`$fieldName`	A logical name of the field to return. Example: `#ew_userfield ("_16_company_name0")`

## ew_userid

`#ew_userid()`

Prints the current user ID to screen; requires no parameters.

ew_url

`#ew_url($url)`

Parameter Name	Description
`$url`	Prints to screen $url with the application url prefix and session data encoded. If you want to make an asset for your portal page available for portal users but not to anyone else, for example an image logo, then you can upload it to the Agiloft server, and use this macro to construct a URL, including session information, to retrieve it.

Example usage:

```
<img border = "0" src = "#ew_url("images/logo.jpg")"/>
```

## ew_view_record

`#ew_view_record($subtypeName $recordId $returnURL $returnFrame)`

Creates a URL which links to the record view.

Parameter Name	Description
`$subtypeName`	Logical name of the table in which the record should be viewed.
`$recordId`	The ID of the record to view.
`$returnURL`	URL to return to after the ticket is created or cancel is pressed

`$returnFrame`	Name of the iframe that contains the view of the table that needs to be refreshed once the operation is completed.

Visualization Parameters ($params)

Visualization parameters apply to the following macros:

- `#ew_table`
- `#ew_searches_list`
- `#ew_query`
- `#ew_new_search`

Parameter Name	Values (default in bold)	Description
`showSearchCriteria`	**true**, false	Whether used search criteria will be shown to user
`workWithPopUp`	true, **false**	Determines whether edit and new operations should be performed witl window
`showToolBar`	**true**, false	Show toolbar on top of table
`showLabel`	**true**, false	Show label for the table
`showNavigation`	**true**, false	Show pages navigation above the table
`showStatus`	**true**, false	Show status of operations on above the table
`showfastsearch`	**true**, false	Show search dialog for the table
`search_context`	true, **false**	Turn on Context for text searches Usage: turn on Context for text searches from the EUI2 by using: `#ew_query("contract" "my_table" "queryId" "showToolBar=false&showNavigation=true&showfastsearch=fa`
`_saved_search_id`	any saved search id	Filter text searches by a saved search Usage: `#ew_query("contract" "my_table" "queryId" "showToolBar=false&showNavigation=true&showfastsearch=fa`

`showLabel`	**true**, false	Show table label above the table or record

Additional Tools

Some helper macros have been implemented to be used in Velocity templates.

User related tools

In a template, the `$ewUser` macros can be used with several methods:

$ewUser.isInGroup (group_name)	Returns true if current user belongs to any of pointed groups. False otherwise. gr_name - names of groups separated by comma. **Example:** `#if ($ewUser.isInGroup("Staff,admin"))` This is text for permitted users only `#else` This is text for all other users #end
$ewUser.isInTeam (team_name)	Returns true if current user belongs to any of specified teams. False otherwise. team_name - names of teams separated by comma
$ewUser. getCurrentUserLogin()	Returns login name of the current user
$ewUser. getCurrentUserFirstName()	Returns the user's First Name, or the user's login if the first name field is empty.
$ewUser. getCurrentUserField(field)	Returns string with the field value from the current user contact. field - name of the contact field. **Example:** Welcome $ewUser.getCurrentUserField("full_name")!

$ewPermission.table

The $ewPermission.table macro describes permission actions for individual users or groups. It returns true when a user has the permission to do the specified action. To add more than one action type, separate each phrase with a comma. For example, `$ewPermission.table ("create", "delete_others")`.

Permission type	Description
Record Permissions	
create	Allow user to create a new record
delete_own	Allow user to delete own records
delete_others	Allow user to delete others’ records
mass_delete	Allow user to delete multiple records
dele_oth_if_pub_y	Delete others’ records if published field is yes
massedit	Allow user to perform mass edit
import_multiple	Allows import of multiple records from a file
export_multiple	Allows export of multiple records to a file
link_multiple_recs	Link multiple records using the Link menu
print_recs	Print records using the printer icon
View and Edit Permissions	
view_own	Allow user to view own records
view_others	Allow user to view others’ records
view_own_if_pub_y	View own records if published field is yes.
view_oth_if_pub_y	View others’ records if published field is yes.
edit_own	Allow user to edit own records
edit_others	Allow user to edit others’ records
Saved Search Permissions	
mod_own_s_searches	Allow user to modify/delete saved searches that they created
mod_all_s_searches	Allow user to modify/delete all saved searches
mod_own_s_views	Allow user to modify/delete saved views that they created
mod_all_s_views	Allow user to modify/delete all saved views
mod_own_s_reports	Allow user to modify/delete saved reports that they created
mod_all_s_reports	Allow user to modify/delete all saved reports
pub_s_reports	Allow user to publish saved reports
v_own_with_ssearch	View own entries that satisfy the saved search

v_oth_with_ssearch	View others’ entries that satisfy the saved search
e_own_with_ssearch	Edit own entries that satisfy the saved search
e_oth_with_ssearch	Edit others’ entries that satisfy the saved search
d_oth_with_ssearch	delete others’ entries that satisfy the saved search
acc_oth_ss	Allow access to others’ saved searches
acc_oth_ss_py	Allow access to others’ saved searches if Published=Yes
FAQ Permissions	
view_faq	Allow user to view FAQs
view_faq_with_ss	View FAQs that satisfy the saved search
Table Permissions	
ed_rec_tab_view	Edit records from table view
omit_table_label	Allow user to set show label on view/editing on General tab
copy_recs	Copy $table-label-plural
show_in_main_tabs	Allow user to see it in main tabs
allow_sel_ed_flds	Allow user to select editable fields in view they can edit
apply_to_subtabls	Apply changes to subtables (in groups wizard)
apply_all_to_subtabls	Apply everything to subtables (in groups wizard)
Custom Report Permissions	
create_calendar	Create calendar records on specific subtype
c_cal_with_ssearch	Create calendars on records that satisfy the saved search
view_own_calendar	View own calendar entries
v_own_cal_with_ssearch	View own entries that satisfy the saved search
view_others_calendar	View other peoples’ calendar entries
others_cal_with_ssearch	View other peoples’ entries that satisfy the saved search
edit_own_calendar	Edit own calendar entries
e_own_cal_with_ssearch	Edit own entries that satisfy the saved search
edit_others_calendar	Edit other peoples’ calendar entries
e_others_cal_with_ssearch	Edit other peoples’ entries that satisfy the saved search

delete_own_calendar	Delete own calendar entries
d_own_cal_with_ssearch	Delete own entries that satisfy the saved search
delete_others_calendar	Delete other peoples’ calendar entries
d_others_cal_with_ssearch	Delete other peoples’ entries that satisfy the saved search
Email Permissions	
view_comm_from	View outgoing email address FROM
edit_comm_from	Edit outgoing email address FROM
view_comm_replyto	View send-back email address REPLY-TO
edit_comm_replyto	Edit send-back email address REPLY-TO
create_comm	Create a communication record linked to the given subtype
Email Template Permissions	
mod_own_eml_tmpl	Allow users to modify/delete own email templates
mod_all_eml_tmpl	Allow users to modify/delete all email templates
viewsend_own_eml_tmpl	Allow viewing/sending only their own email templates
viewsend_ownpublished_eml_tmpl	Allow viewing/sending own and published email templates
viewsend_ownpublished_eml_tmpl	Allow viewing/sending own and published email templates
viewsend_all_eml_tmpl	Allow viewing/sending of all email templates
publish_eml_tmpl	Allow user to publish email templates
Print Template Permissions	
cr_mod_own_ptmpl	Create/modify own print templates
cr_mod_all_ptmpl	Create/modify all print templates
use_all_ptmpl	Use all print templates
use_ownpub_ptmpl	Use/own/publish print templates
can_pub_ptmpl	User can publish print templates

Response related tools

The `$ewResponse` tool can be used with several methods in a template:

Method name	Description
$ewResponse.nocache()	Set response header not to cache the template
$ewResponse.cache(time)	Set response header to cache content for *time* minutes

Localization related tools

Templates are designed with national language support. The text displayed to users can be maintained in separate language-based templates and displayed based on the locale. There are predefined templates with names such as `translation.properties` that contain localized text in the form

key1=value1

key2=value2

If you want to translate your templates to another language, you should create a template named `translation_<language-code>.properties` and insert your translations there. If a key is not found in the localized properties template, the text from the default `translation.properties` file is used.

Method name	Description
$ewText.get ($key)	Reads the current session locale, then looks up the key line in the corresponding `translation.properties` file and prints the text value to the screen. For example: `Welcome To $ewText.get("home.sc.title")!`
`$ewText.get (key)`	Returns localized message
`$ewText.get (key, [value1, value2,...])`	Returns localized message with values substituted

`$text.<GUI2key>`	This method returns values from the system-wide properties file rather than the EUI Template. Use the key name from the file GUI2.properties in the Agiloft installation directory. For instructions to download the system properties file, see the Localization topic. For example, `$text.eui.name.text1` displays 'User:' when used in an EUI template.

Knowledgebase related tools

The `$ewUtils` tool returns name and label information about the knowledgebase.

Method	Description
$ewUtils.thisProjectLabel	Returns the knowledgebase label when included in a template.
$ewUtils.thisProjectName	Returns the knowledgebase name when included in a template.