



Workflow Designer

Template Field Features

Allmost all tasks in Workflow configuration are template fields and could be filled with variables and expressions.

All textfields which look like this, are capable of this feature:



If the blue Icon is green, then the field is not a default Template field, but a Expression field. The **MUST** contain a valid PHP script and return a value by “**return**” statement.

You could use the following features within this textfields:

Default variables

This is the most common way to use templates and is available in both field types.

The variable name is the name of the field you want to access.

This variables will be replaced by the value of this field, from record, which runs the workflow. For example there are variables like **\$accountname**, **\$lastname**, **\$firstname**, **\$bill_street**,

A list of all possible variables, you get by click on the blue/green box icon next to the textfield. This variables could be used inside the default blue template fields and the “green” textareas.

This is a readonly Access, which couldn't be used to change the content of a field.

Relation fields

Format in default blue template fields:

```
$(contact_id: (Contacts) leadsource)
```

Format in expression green template fields:

```
<?php ... custom function ... return "content"; ?>
```

The format differ, because the green template fields must be valid PHP code and the structure with brackets is known from other modules, too.

This could be used to access the value of a field inside a referenced record. (For example a value from Organization in Contacts module)

All possible variables (also this relations) are also listed over the blue/green icon. *You don't need to create the structure by yourself!*

In the blue Textbox this variable use this structure: **\$(localFieldname: (relatedModuleName) relatedRecordFieldname)**

The green textbox use another structure, because of the php base: **\$localFieldname→relatedModuleName→relatedRecordFieldname**

Shortcuts

There are some special shortcut functions available within the default template field, which allows you to do simple manipulations.

Shortcut	Description
<code>\$(nl2br,\$fieldvalue)</code>	Should be used to output any fields, texts or anything else, filled by a user, within E-Mail Body. Reason for this is the difference within newline command. The default carriage return of textfields will be ignored in HTML E-Mails. They need a <code>
</code> HTML for newline
<code>\$(Now)</code>	insert the current date in format (YYYY-MM-DD)
<code>\$(Now,-x)</code>	insert the current date -X days in format (YYYY-MM-DD)
<code>\$(Now,+x)</code>	insert the current date +X days in format (YYYY-MM-DD)
<code>\$(Now,+x days/weeks/months)</code>	insert the current date with modifications from interval (YYYY-MM-DD) (Available interval settings: <code>http://php.net/manual/de/datetime.formats.relative.php</code>)
<code>\$(Now,0,'d.m.Y')</code>	insert the current date in format (DD.MM.YYYY) <code>You get possible placeholders here.</code>
<code>\$(Now,0,'H:i:s')</code>	insert the current time in format (HH:MM:SS) <code>You get possible placeholders here.</code>
<code>\$(Now,"+5 days",,\$now)</code>	(From version 600.1701) - Date calculations are relative to the date within \$now variable. Could be static string or fieldvariable. Third format parameter could also be set!
<code>\$(entityname, \$id)</code>	Replace the ID with the Name of the Record (Accountname, Firstname & Lastname, ...)
<code>\$(dateformat,\$date,"d.m.Y")</code>	insert the given datetime in format (DD.MM.YYYY) <code>You get possible placeholders here.</code>
<code>\$(utctz,\$time)</code>	If you create events and want to use a time field as source for the start time you probably need this function to clear the timezone from the time field.
<code>\$(link,\$crmid)</code>	Insert a HTML Link with Record Label
<code>\$(round,\$number,\$decimals)</code>	Round a number to \$decimals places
<code>\$(number,\$value)</code>	Convert second parameter to number and strip all unnecessary decimal places
<code>\$(number,\$value,\$decimals)</code>	Convert second parameter to number, but keep given decimals
<code>\$(timezone,\$timefield,"Europe/Berlin")</code>	Convert a given time into any Timezone. Useful to output times from Calendar
<code>\$(currency,\$value)</code>	Convert a number into configured currency format of vtiger

If you have basic PHP skills, you are able to add your own shortcut functions, you could use. Go to developer documentation

Variable modifiers

Sometimes you only need a simple modifier for your variables. For example change date format or execute some simple PHP functions, like strtolower, ucfirst, ... on a variable.

For this we follow the way, the Smarty template engine goes and implement "modifiers". To use them is as easy as to use simple variables.

For example if you include a date from field, you normally use **\$datefield**. If you use a modified you surround the variable with {..} and add the modified by using a pipe.

For example you could use the datefield like this: **{ \$datefield|date_format:"d.m.Y" }** to output the date in format d.m.Y. You could use every placeholder from this

list: <http://php.net/manual/de/function.date.php>

You could use every Smarty modifier and all functions, which are allowed in custom expressions, too. (See the list on top of this page: [type the link label](#))

Translations

Template fields can use translations of strings, which are loaded from module language file of current Record.

To use this feature surround the string to translate, by % .

For example to translate the word "from", write: **%from%**

Please read the following sections only, if you have some basic development skills, because they make use of the custom functions, which must follow special rules.

Custom functions inside default template fields

A big improvement was the feature to insert a custom expression in every default template field (with blue icon).

This makes it possible to generate very complex and individual values by the the usage of php functions, without the limitations of simple template fields.

You only need to insert the following:

```
<?php ... custom function ... return "content"; ?>
```

Only within the HTML Editor of "send mail" Task, these tags don't work, because they are removed by editor. Then you need to use **`#{ ... }>`** instead of **`<?php ... ?>`** and could use this feature.

The return statement is imperative, because the complete block (from "`<?php`" to "`?>`") will be replaced by the return value.

This allows for example the insert of a rounded total sum of an invoice by:

```
<?php return round($hdnGrandTotal, 2); ?>
```

Also this feature allows the calculation of dates from fields by:

```
<?php $date = wf_date$fieldname, "+1 days"); ?>
```

You also have access to the Environmental variable within this tags.

*Please note, you have to use the related variable structure of custom functions (**`$field→module→field`**) and not the Template structure of variables. (`$field (module) field`) The Variable chooser do not know this and you need to change the format for yourself.*

Other examples I will mention on a the custom function page.

